```
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBB         BBB  AAA         AAA  CCC            KKK         KKK  UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK         KKK  UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK         KKK  UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK       KKK    UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK       KKK    UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK       KKK    UUU          UUU  PPP         PPP
BBBBBBBBBBBB     AAA         AAA  CCC            KKKKKKKKK        UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB     AAA         AAA  CCC            KKKKKKKKK        UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB     AAA         AAA  CCC            KKKKKKKKK        UUU          UUU  PPPPPPPPPPPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC            KKK       KKK    UUU          UUU  PPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC            KKK       KKK    UUU          UUU  PPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC            KKK       KKK    UUU          UUU  PPP
BBB         BBB  AAA         AAA  CCC            KKK          KKK  UUU          UUU  PPP
BBB         BBB  AAA         AAA  CCC            KKK          KKK  UUU          UUU  PPP
BBB         BBB  AAA         AAA  CCC            KKK          KKK  UUU          UUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCC  KKK          KKK  UUUUUUUUUUUUUUUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCC  KKK          KKK  UUUUUUUUUUUUUUUU  PPP
```

```
  SSSSSSSS  TTTTTTTTTT    AAAAAA      AAAAAA      CCCCCCCC  PPPPPPPP
  SSSSSSSS  TTTTTTTTTT    AAAAAA      AAAAAA      CCCCCCCC  PPPPPPPP
SS              TT       AA    AA    AA    AA    CC        PP      PP
SS              TT       AA    AA    AA    AA    CC        PP      PP
SS              TT       AA    AA    AA    AA    CC        PP      PP
SS              TT       AA    AA    AA    AA    CC        PP      PP
  SSSSSS        TT       AA    AA    AA    AA    CC        PPPPPPPP
  SSSSSS        TT       AA    AA    AA    AA    CC        PPPPPPPP
        SS      TT       AAAAAAAAAA  AAAAAAAAAA  CC        PP
        SS      TT       AAAAAAAAAA  AAAAAAAAAA  CC        PP
        SS      TT       AA    AA    AA    AA    CC        PP              ....
        SS      TT       AA    AA    AA    AA    CC        PP              ....
SSSSSSSS        TT       AA    AA    AA    AA      CCCCCCCC  PP            ....
SSSSSSSS        TT       AA    AA    AA    AA      CCCCCCCC  PP            ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;     1        0001  0 MODULE STAACP   (%TITLE 'Standalone ACP'
;     2        0002  0                  IDENT = 'V04-000'
;     3        0003  0                  ) =
;     4        0004  1 BEGIN
;     5        0005  1
;     6        0006  1 !
;     7        0007  1 !**************************************************************************
;     8        0008  1 !*                                                                        *
;     9        0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    10        0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    11        0011  1 !*  ALL RIGHTS RESERVED.                                                   *
;    12        0012  1 !*                                                                        *
;    13        0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    14        0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    15        0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    16        0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    17        0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    18        0018  1 !*  TRANSFERRED.                                                           *
;    19        0019  1 !*                                                                        *
;    20        0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    21        0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    22        0022  1 !*  CORPORATION.                                                           *
;    23        0023  1 !*                                                                        *
;    24        0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    25        0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;    26        0026  1 !*                                                                        *
;    27        0027  1 !*                                                                        *
;    28        0028  1 !**************************************************************************
;    29        0029  1 !
;    30        0030  1
;    31        0031  1 !++
;    32        0032  1 ! FACILITY:
;    33        0033  1 !       Backup/Restore
;    34        0034  1 !
;    35        0035  1 ! ABSTRACT:
;    36        0036  1 !       This module contains the standalone ACP routines.
;    37        0037  1 !
;    38        0038  1 ! ENVIRONMENT:
;    39        0039  1 !       VAX/VMS user mode.
;    40        0040  1 !--
;    41        0041  1 !
;    42        0042  1 ! AUTHOR: M. Jack, CREATION DATE: 01-Feb-1981
;    43        0043  1 !
;    44        0044  1 ! MODIFIED BY:
;    45        0045  1 !
;    46        0046  1 !       V03-019 LMP0301         L. Mark Pilant,         10-Aug-1984  8:34
;    47        0047  1 !               Fix a bug from LMP0272 that caused BACKUP to ACCVIO during
;    48        0048  1 !               an ACCESS.
;    49        0049  1 !
;    50        0050  1 !       V03-018 LY0516          Larry Yetto             25-JUL-1984 15:27
;    51        0051  1 !               Zero fill the FIB in STA_ENTER so that sequential disk save sets
;    52        0052  1 !               will work again.
;    53        0053  1 !
;    54        0054  1 !       V03-017 LMP0272         L. Mark Pilant,         3-Jul-1984  10:37
;    55        0055  1 !               Add the FIB as an argument for the ACL processing.
;    56        0056  1 !
;    57        0057  1 !       V03-016 ACG0415         Andrew C. Goldstein,    24-Apr-1984  18:18
```

```
;    58          0058  1 !               Fix boundary bugs in ACL handling
;    59          0059  1 !
;    60          0060  1 !       V03-015 ACG0382         Andrew C. Goldstein,    16-Dec-1983  16:49
;    61          0061  1 !               Fix RVN usage in ACL processing; add validation for
;    62          0062  1 !               VCB_INIT_DONE in STA_CREATE and STA_DISMOUNT_OUTPUT.
;    63          0063  1 !               Add error cleanup in STA_ACCESS.
;    64          0064  1 !
;    65          0065  1 !       V03-014 ACG0365         Andrew C. Goldstein,    11-Oct-1983  14:48
;    66          0066  1 !               Tie off ACL processing in BACKUP sequential disk
;    67          0067  1 !
;    68          0068  1 !       V03-013 ACG53087        Andrew C. Goldstein,    30-Aug-1983  19:28
;    69          0069  1 !               Fix creation of save sets at end of MFD in seq disk
;    70          0070  1 !
;    71          0071  1 !       V03-012 ACG0352         Andrew C. Goldstein,    22-Aug-1983  17:46
;    72          0072  1 !               Fix descriptor initialization bugs introduced in LMP0118;
;    73          0073  1 !               fix bug in mounting sequential disk volumes for input
;    74          0074  1 !
;    75          0075  1 !       V03-011 LMP0118         L. Mark Pilant,          9-Jun-1983  11:01
;    76          0076  1 !               Correct problems with trying to create a file whose ACL
;    77          0077  1 !               spans headers.
;    78          0078  1 !
;    79          0079  1 !       V03-010 ACG0332         Andrew C. Goldstein,    20-Apr-1983  17:58
;    80          0080  1 !               Add support for file highwater mark and RMS journal flags
;    81          0081  1 !
;    82          0082  1 !       V03-009 ACG0334         Andrew C. Goldstein,    6-May-1983  14:39
;    83          0083  1 !               Fix inconsistencies in declaration of FILE_ERROR
;    84          0084  1 !
;    85          0085  1 !       V03-008 ACG0325         Andrew C. Goldstein,    4-Apr-1983  15:52
;    86          0086  1 !               Fix file header area length validation, add extended
;    87          0087  1 !               file name support.
;    88          0088  1 !
;    89          0089  1 !       V03-007 LMP0085         L. Mark Pilant,          2-Mar-1983  15:34
;    90          0090  1 !               Fix a problem that caused the file to remain open after
;    91          0091  1 !               giving an error about not being able to write the entire
;    92          0092  1 !               ACL.
;    93          0093  1 !
;    94          0094  1 !       V03-006 ACG0313         Andrew C. Goldstein,    11-Feb-1983  1:16
;    95          0095  1 !               Fix accumulation of blocks returned in STA_EXTEND.
;    96          0096  1 !               Also remove zeroing of bitmap of mounted disks.
;    97          0097  1 !
;    98          0098  1 !       V03-005 LMP0067         L. Mark Pilant,         15-Dec-1982  15:12
;    99          0099  1 !               Deallocate memory obtained for ACL segment storage when the
;   100          0100  1 !               file is deaccessed.
;   101          0101  1 !
;   102          0102  1 !       V03-004 LMP0044         L. Mark Pilant,          3-Nov-1982  10:20
;   103          0103  1 !               Add support for saving and restoring ACL's.
;   104          0104  1 !
;   105          0105  1 !       V03-003 MLJ0100         Martin L. Jack, 7-Oct-1982  15:04
;   106          0106  1 !               In write attributes, add range checking for length of ident
;   107          0107  1 !               area.
;   108          0108  1 !
;   109          0109  1 !       V03-002 ACG0281         Andrew C. Goldstein,     5-Apr-1982  16:02
;   110          0110  1 !               Add ODS-1 multi-header index file support
;   111          0111  1 !
;   112          0112  1 !       V03-001 ACG0279         Andrew C. Goldstein,     1-Apr-1982  14:20
;   113          0113  1 !               Rework header processing in STA_EXTEND
;   114          0114  1 !
```

```
;  115     0115  1 ;       V02-009 MLJ0081         Martin L. Jack, 26-Feb-1982  16:03
;  116     0116  1 ;               Implement RETAINMIN and RETAINMAX for new home block fields.
;  117     0117  1 ;
;  118     0118  1 ;       V02-008 MLJ0075         Martin L. Jack, 31-Jan-1982  7:26
;  119     0119  1 ;               Correct access violation introduced in V02-007.
;  120     0120  1 ;
;  121     0121  1 ;       V02-007 MLJ0062         Martin L. Jack, 10-Dec-1981  20:00
;  122     0122  1 ;               Rework STA_ACCESS to allow IO$_ACCESS without IO$M_ACCESS when
;  123     0123  1 ;               file is already accessed.  This is necessary to avoid spurious
;  124     0124  1 ;               FILALRACC errors when saving multi-header files.
;  125     0125  1 ;
;  126     0126  1 ;       V02-006 ACG0236         Andrew C. Goldstein,    8-Dec-1981  21:45
;  127     0127  1 ;               Check status from PACKACK function
;  128     0128  1 ;
;  129     0129  1 ;       V02-005 MLJ0054         Martin L. Jack, 20-Oct-1981  8:30
;  130     0130  1 ;               Implement /VOLUME.  Reconstruct quota file for /IMAGE output.
;  131     0131  1 ;               Release disk space on a failed IO$_CREATE.  Display pertinent
;  132     0132  1 ;               file name in error messages issued by STA_DISMOUNT_OUTPUT and
;  133     0133  1 ;               STA_INIT_HDRS.  Move globals to common.  Integrate GET_VM and
;  134     0134  1 ;               FREE_VM jacket routines.
;  135     0135  1 ;
;  136     0136  1 ;       V02-004 MLJ0040         Martin L. Jack, 3-Sep-1981  19:42
;  137     0137  1 ;               Clean up window on a failed IO$_CREATE to avoid
;  138     0138  1 ;               incorrect "file already accessed" errors.
;  139     0139  1 ;
;  140     0140  1 ;       V02-003 MLJ0039         Martin L. Jack, 3-Sep-1981  19:21
;  141     0141  1 ;               Include general-mode addressing where required.
;  142     0142  1 ;
;  143     0143  1 ;       V02-002 ACG0211         Andrew C. Goldstein,    22-Jul-1981  17:18
;  144     0144  1 ;               Add logic to create save set files
;  145     0145  1 ;
;  146     0146  1 ;       V02-001 MLJ0025         Martin L. Jack, 8-May-1981  14:24
;  147     0147  1 ;               Move setting of index file bitmap into WRITE_HEADER.  Improve
;  148     0148  1 ;               documentation.  Avoid creating window for IO$_ACCESS (and
;  149     0149  1 ;               disallow STATBLK attribute) unless file is actually accessed.
;  150     0150  1 ;               Do preliminary work for file placement.
;  151     0151  1 ;
;  152     0152  1 ;**
```

```
;  154          0153  1 REQUIRE 'SRC$:COMMON';
;  155          1259  1 LIBRARY 'SYS$LIBRARY:LIB';
;  156          1260  1 REQUIRE 'LIB$:BACKDEF';
;  157          1710  1
;  158          1711  1
;  159          1712  1 LINKAGE
;  160          1713  1         L_DQF_WRITE_ENTRY = CALL:
;  161          1714  1             GLOBAL(
;  162          1715  1                 DQF_BUFFER=11,
;  163          1716  1                 DQF_RECORD=10,
;  164          1717  1                 DQF_VBN=9,
;  165          1718  1                 DQF_DEFAULT_PERM=8,
;  166          1719  1                 DQF_DEFAULT_OVER=7);
;  167          1720  1
;  168          1721  1
;  169          1722  1 FORWARD ROUTINE
;  170          1723  1         DQF_FIND_UIC,                      ! Search quota table for UIC
;  171          1724  1         DQF_MODIFY_USAGE:
;  172          1725  1                         NOVALUE,           ! Update usage data for UIC
;  173          1726  1         DQF_WRITE_ENTRY:L_DQF_WRITE_ENTRY NOVALUE,
;  174          1727  1                                            ! Rewrite quota file entry
;  175          1728  1         ASSIGN_INPUT_CHANNEL,              ! Assign a channel to input disk
;  176          1729  1         ASSIGN_OUTPUT_CHANNEL,             ! Assign a channel to output disk
;  177          1730  1         SWITCH_VOLUME,                     ! Switch to relative volume
;  178          1731  1         VERIFY_HEADER,                     ! Check one file header
;  179          1732  1         READ_HEADER,                       ! Read one file header
;  180          1733  1         WRITE_HEADER,                      ! Write one file header
;  181          1734  1         CREATE_DELHDR:  NOVALUE,           ! Create deleted file header
;  182          1735  1         TAKE_BLOCKS:    NOVALUE,           ! Take blocks from free list
;  183          1736  1         STA_ALLOC_LBN,                     ! Allocate specific LBN
;  184          1737  1         STA_ALLOC_BEST,                    ! Allocate best fit
;  185          1738  1         FREE_BLOCKS:    NOVALUE,           ! Return blocks to free list
;  186          1739  1         MAKE_POINTER1,                     ! Make map pointer (ODS-1)
;  187          1740  1         MAKE_POINTER,                      ! Make map pointer (ODS-2)
;  188          1741  1         CREATE_WINDOW,                     ! Create window block
;  189          1742  1         DELETE_WINDOW:  NOVALUE,           ! Delete window block
;  190          1743  1         ADD_BLACKHOLE_MAP:
;  191          1744  1                         NOVALUE,           ! Add blackhole pointer to window
;  192          1745  1         ADD_WINDOW_MAP: NOVALUE,           ! Add map entry to window
;  193          1746  1         QIO_AST:        NOVALUE,           ! Completion AST for R_W_VIRTUAL
;  194          1747  1         R_W_VIRTUAL,                       ! Read/write virtual
;  195          1748  1         STA_INIVOL:     NOVALUE,           ! Initialize a volume
;  196          1749  1         STA_INIT_HDRS:  NOVALUE,           ! Initialize file headers
;  197          1750  1         STA_WRITEBOOT:  NOVALUE,           ! Rewrite boot block with boot LBN
;  198          1751  1         STA_MOUNT:      NOVALUE,           ! Mount a volume set
;  199          1752  1         READY_DISK,                        ! Ready disk for save set
;  200          1753  1         STA_ENTER:      NOVALUE,           ! Execute RMS ENTER function
;  201          1754  1         STA_EXTEND,                        ! Incrementally extend file
;  202          1755  1         STA_RDWRVBLK,                      ! Execute IO$_READVBLK, IO$_WRITEVBLK
;  203          1756  1         STA_ACCESS,                        ! Execute IO$_ACCESS
;  204          1757  1         CREATE_CLEANUP,                    ! Process allocation list after create
;  205          1758  1         CREATE_EXTHDR,                     ! Create extension header
;  206          1759  1         STA_CREATE,                        ! Execute IO$_CREATE
;  207          1760  1         STA_DEACCESS,                      ! Execute IO$_DEACCESS
;  208          1761  1         STA_MODIFY,                        ! Execute IO$_MODIFY
;  209          1762  1         STA_QIO,                           ! Dispatch ACP QIOs
;  210          1763  1         STA_QIOW,                          ! Execute $QIOW
```

```
;  211          1764  1          READ_ATTRIBUTES,                    ! Read attributes
;  212          1765  1          TO_ODS1_DATE:   NOVALUE,             ! Convert 64-bit time to ODS-1 format
;  213          1766  1          WRITE_ATTRIBUTES;                   ! Write attributes
;  214          1767  1
;  215          1768  1
;  216          1769  1 EXTERNAL ROUTINE
;  217          1770  1          CHECKSUM,                           ! Compute file header checksum
;  218          1771  1          CHECKSUM2,                          ! Compute home block checksum
;  219          1772  1          FILE_ERROR:     NOVALUE,            ! Signal file-related error
;  220          1773  1          FROM_ODS1_DATE: NOVALUE,            ! Convert ODS-1 format to 64-bit time
;  221          1774  1          INITIALIZE_VOLUME:
;  222          1775  1                          NOVALUE,            ! Execute volume initialization
;  223          1776  1          MAKE_NAMEBLOCK: NOVALUE,            ! Convert filename to ODS-1 format
;  224          1777  1          FREE_VM:        NOVALUE,            ! Deallocate virtual memory
;  225          1778  1          GET_VM,                             ! Allocate virtual memory
;  226          1779  1          GET_ZERO_VM,                        ! Allocate and clear virtual memory
;  227          1780  1          SYS$ASSIGN:     ADDRESSING_MODE(GENERAL),
;  228          1781  1                                              ! Assign channel system service
;  229          1782  1          ACL_DELETEACL,                      ! Delete and deallocate ACL segments
;  230          1783  1          ACL_BUILDACL,                       ! Build the ACL
;  231          1784  1          ACL_DISPATCH;                       ! ACL function dispatcher
;  232          1785  1
;  233          1786  1
;  234          1787  1 EXTERNAL LITERAL
;  235          1788  1          BACKUP$_BADSETCNT,
;  236          1789  1          BACKUP$_DISKFRAG,
;  237          1790  1          BACKUP$_GETCHN,
;  238          1791  1          BACKUP$_INCRVN,
;  239          1792  1          BACKUP$_INCSETCNT,
;  240          1793  1          BACKUP$_INVATTVAL,
;  241          1794  1          BACKUP$_INVFID,
;  242          1795  1          BACKUP$_INVHOMBLK,
;  243          1796  1          BACKUP$_LARGECNT,
;  244          1797  1          BACKUP$_NOBITMAP,
;  245          1798  1          BACKUP$_NOHOMEBLK,
;  246          1799  1          BACKUP$_NOINDEXF,
;  247          1800  1          BACKUP$_NOVOLDATA,
;  248          1801  1          BACKUP$_ODS2SAVE,
;  249          1802  1          BACKUP$_OPENIN,
;  250          1803  1          BACKUP$_OPENOUT,
;  251          1804  1          BACKUP$_CLOSEOUT,
;  252          1805  1          BACKUP$_READBMAP,
;  253          1806  1          BACKUP$_READERR,
;  254          1807  1          BACKUP$_READIMAP,
;  255          1808  1          BACKUP$_READYREAD,
;  256          1809  1          BACKUP$_READYWRITE,
;  257          1810  1          BACKUP$_STRUCLEV,
;  258          1811  1          BACKUP$_VOLINSET,
;  259          1812  1          BACKUP$_WRITENABLE,
;  260          1813  1          BACKUP$_WRITEERR,
;  261          1814  1          BACKUP$_QUOTAFILE;
;  262          1815  1
;  263          1816  1
;  264          1817  1 G$DEFINE();                  ! Define global common area
;  265          1818  1
;  266          1819  1
;  267          1820  1 OWN
```

```
;  268          1821  1          RSA_DESC:       VECTOR[2],          ! Descriptor for RSA in STA_DISMOUNT_OUTPUT
;  269          1822  1          DQF_QUOTA_FID:  BBLOCK[FID$C_LENGTH], ! File ID of quota file
;  270          1823  1          DQF_ROOT,                           ! Root of quota table
;  271          1824  1          DQF_COUNT,                          ! Count of entries in quota table
;  272          1825  1          QUEUE_HEADERS:  VECTOR [6];         ! Queue headers for:
;  273          1826  1                                              !     Allocated disk extents
;  274          1827  1                                              !     Required extents
;  275          1828  1                                              !     Used extension file IDs
;  276          1829  1
;  277          1830  1
;  278          1831  1 MACRO
;  279          1832  1
;  280          1833  1          ! Field definitions for extent list.
;  281          1834  1          !
;  282          1835  1          EXT_FLINK=      0,0,32,0 %,         ! Forward link
;  283          1836  1          EXT_BLINK=      4,0,32,0 %,         ! Backward link
;  284          1837  1          EXT_VCB=        8,0,32,0 %,         ! Pointer to VCB for volume
;  285          1838  1          EXT_COUNT=      12,0,32,0 %,        ! Count of blocks
;  286          1839  1          EXT_LBN=        16,0,32,0 %;        ! LBN of blocks
;  287          1840  1
;  288          1841  1 LITERAL
;  289          1842  1          EXT_S_ENTRY=    20;                 ! Size of extent list entry in bytes
;  290          1843  1
;  291          1844  1
;  292          1845  1 MACRO
;  293          1846  1
;  294          1847  1          ! Field definitions for create list.
;  295          1848  1          !
;  296          1849  1          CRT_FLINK=      0,0,32,0 %,         ! Forward link
;  297          1850  1          CRT_BLINK=      4,0,32,0 %,         ! Backward link
;  298          1851  1
;  299          1852  1          CRT_FID_FQHDR=  8,0,32,0 %,         ! Queue header for FID queue
;  300          1853  1          CRT_FID_BQHDR=  12,0,32,0 %,        !
;  301          1854  1          CRT_BLOCKS=     16,0,32,0 %,        ! Blocks
;  302          1855  1
;  303          1856  1          CRT_FID=        8,0,0,0 %,          ! File ID
;  304          1857  1          CRT_FID_NUM=    8,0,16,0 %,         !
;  305          1858  1          CRT_FID_SEQ=    10,0,16,0 %,        !
;  306          1859  1          CRT_FID_RVNU=   12,0,16,0 %,        !
;  307          1860  1          CRT_FID_RVN=    12,0,8,0 %,         !
;  308          1861  1          CRT_FID_NMX=    13,0,8,0 %;         !
;  309          1862  1
;  310          1863  1
;  311          1864  1 LITERAL
;  312          1865  1          CRT_S_BLOCKS=   20,                 ! Size of "blocks" entry
;  313          1866  1          CRT_S_FID=      14;                 ! Size of "FID" entry
;  314          1867  1
;  315          1868  1
;  316          1869  1 BUILTIN
;  317          1870  1          CALLG,
;  318          1871  1          INSQUE,
;  319          1872  1          REMQUE,
;  320          1873  1          TESTBITSC,
;  321          1874  1          ROT;
```

```
;  323     1875  1 %SBTTL 'DQF_FIND_UIC - find UIC in quota table'
;  324     1876  1 ROUTINE DQF_FIND_UIC (UIC)=
;  325     1877  1
;  326     1878  1 !++
;  327     1879  1 !
;  328     1880  1 ! FUNCTIONAL DESCRIPTION:
;  329     1881  1 !       This routine finds the quota table entry for a specified UIC,
;  330     1882  1 !       creating it if necessary.
;  331     1883  1 !
;  332     1884  1 ! INPUT PARAMETERS:
;  333     1885  1 !       UIC             - The UIC.
;  334     1886  1 !
;  335     1887  1 ! IMPLICIT INPUTS:
;  336     1888  1 !       DQF_ROOT        - The root of the quota table structure.
;  337     1889  1 !
;  338     1890  1 ! OUTPUT PARAMETERS:
;  339     1891  1 !       NONE
;  340     1892  1 !
;  341     1893  1 ! IMPLICIT OUTPUTS:
;  342     1894  1 !       NONE
;  343     1895  1 !
;  344     1896  1 ! ROUTINE VALUE:
;  345     1897  1 !       A pointer to the entry for the specified UIC.
;  346     1898  1 !
;  347     1899  1 ! SIDE EFFECTS:
;  348     1900  1 !       The entry may be created if required.
;  349     1901  1 !
;  350     1902  1 !--
;  351     1903  1
;  352     1904  2 BEGIN
;  353     1905  2 LOCAL
;  354     1906  2         P:              REF BBLOCK,     ! Pointer to DQF entry
;  355     1907  2         Q:              REF BBLOCK;     ! Pointer to link to DQF entry
;  356     1908  2
;  357     1909  2
;  358     1910  2 ! Search DQF table for a matching entry or to find where the new entry
;  359     1911  2 ! must be inserted.
;  360     1912  2 !
;  361     1913  2 Q = DQF_ROOT;                           ! Point to root of table
;  362     1914  2 P = ..Q;                                ! Point to highest entry in table
;  363     1915  2 WHILE .P NEQ 0 DO                       ! Until bottom of table reached
;  364     1916  3     BEGIN
;  365     1917  3     IF .UIC EQL .P[DQF_UIC]             ! If correct entry
;  366     1918  3         THEN RETURN .P;                 !     return it
;  367     1919  3     IF .UIC GTRU .P[DQF_UIC]            ! If desired entry is on right branch
;  368     1920  3         THEN Q = P[DQF_RLINK]           !     point to right link word
;  369     1921  3         ELSE Q = P[DQF_LLINK];          !     otherwise to left link word
;  370     1922  3     P = ..Q;                            ! Point to right or left branch
;  371     1923  2     END;
;  372     1924  2
;  373     1925  2
;  374     1926  2 ! Allocate and initialize the new entry.
;  375     1927  2 !
;  376     1928  2 DQF_COUNT = .DQF_COUNT + 1;
;  377     1929  2 .Q = GET_VM(DQF_S_ENTRY);
;  378     1930  2 P = ..Q;
;  379     1931  2 P[DQF_LLINK] = 0;
```

```
;  380          1932  2 P[DQF_RLINK] = 0;
;  381          1933  2 P[DQF_UIC] = .UIC;
;  382          1934  2 P[DQF_USAGE] = 0;
;  383          1935  2 P[DQF_PERMQUOTA] = -1;                ! Flag that no quotas exist
;  384          1936  2 P[DQF_OVERDRAFT] = -1;
;  385          1937  2 .P
;  386          1938  1 END;


                                .TITLE  STAACP Standalone ACP
                                .IDENT  \V04-000\

                                .PSECT  COMMON,NOEXE,  OVR,2

00000 GLOBAL_BASE:
                                .BLKB   0
00000 FREE_LIST:
                                .BLKB   8
00008 INPUT_WAIT:
                                .BLKB   8
00010 REREAD_WAIT:
                                .BLKB   8
00018 OUTPUT_WAIT:
                                .BLKB   8
00020 JPI_UIC:.BLKB   4
00024 JPI_USERNAME:
                                .BLKB   12
00030 JPI_DATE:
                                .BLKB   8
00038 JPI_NODE_DESC:
                                .BLKB   8
00040 JPI_CURPRIV:
                                .BLKB   8
00048 SYI_VERSION:
                                .BLKB   4
0004C SYI_SID:.BLKB   4
00050 RWSV_HOLD_LIST:
                                .BLKB   8
00058 RWSV_CRC16:
                                .BLKB   64
00098 RWSV_AUTODIN:
                                .BLKB   64
000D8 RWSV_FILESET_ID:
                                .BLKB   8
000E0 RWSV_VOLUME_ID:
                                .BLKB   12
000EC RWSV_VOL_NUMBER:
                                .BLKB   2
000EE RWSV_SEG_NUMBER:
                                .BLKB   2
000F0 RWSV_FILE_NUMBER:
                                .BLKB   4
000F4 RWSV_SAVE_QUAL:
                                .BLKB   4
000F8 RWSV_SAVE_FAB:
                                .BLKB   4
000FC RWSV_CHAN:
```

```
                        .BLKB   4
00100 RWSV_XOR_BCB:
                        .BLKB   4
00104 RWSV_IN_SEQ:
                        .BLKB   4
00108 RWSV_IN_SEQ_0:
                        .BLKB   4
0010C RWSV_IN_XOR_SEQ:
                        .BLKB   4
00110 RWSV_IN_XOR_RFA:
                        .BLKB   6
00116 RWSV_LOOKAHEAD:
                        .BLKB   1
00117 RWSV_XORSIZE:
                        .BLKB   1
00118 RWSV_IN_GROUP_SIZE:
                        .BLKB   4
0011C RWSV_IN_ERRORS:
                        .BLKB   2
0011E RWSV_IN_XORUSE:
                        .BLKB   2
00120 RWSV_IN_ORGERR:
                        .BLKB   8
00128 RWSV_IN_VBN:
                        .BLKB   4
0012C RWSV_IN_VBN_0:
                        .BLKB   4
00130 RWSV_ALLOC:
                        .BLKB   4
00134 RWSV_EOF:
                        .BLKB   4
00138 RWSV_OUT_SEQ:
                        .BLKB   4
0013C RWSV_OUT_VBN:
                        .BLKB   4
00140 RWSV_OUT_BLOCK_COUNT:
                        .BLKB   4
00144 RWSV_OUT_ERRORS:
                        .BLKB   2
00146 RWSV_SEQ_ERRORS:
                        .BLKB   2
00148 RWSV_OUT_GROUP_COUNT:
                        .BLKB   1
00149 RWSV_PADDING:
                        .BLKB   3
0014C QUAL:             .BLKB   112
001BC COM_SSNAME:
                        .BLKB   8
001C4 COM_VALID_TYPES:
                        .BLKB   2
001C6 COM_FLAGS:
                        .BLKB   2
001C8 COM_PADDING:
                        .BLKB   1
001C9 COM_BUFF_COUNT:
                        .BLKB   1
001CA COM_I_SETCOUNT:
```

```
                        .BLKB   1
001CB COM_O_SETCOUNT:
                        .BLKB   1
001CC COM_I_STRUCNAME:
                        .BLKB   12
001D8 COM_O_STRUCNAME:
                        .BLKB   12
001E4 COM_O_BSRDATE:
                        .BLKB   8
001EC ALT_SSNAME:
                        .BLKB   32
0020C INPUT_FUNC:
                        .BLKB   1
0020D INPUT_RTYPE:
                        .BLKB   1
0020E OUTPUT_FUNC:
                        .BLKB   1
0020F FAST_STRUCLEV:
                        .BLKB   1
00210 INPUT_BEG:
                        .BLKB   0
00210 INPUT_CHAN:
                        .BLKB   4
00214 INPUT_FLAGS:
                        .BLKB   2
00216 INPUT_PADDING:
                        .BLKB   2
00218 INPUT_FAB:
                        .BLKB   4
0021C INPUT_NAM:
                        .BLKB   4
00220 INPUT_BCB:
                        .BLKB   4
00224 INPUT_QUAL:
                        .BLKB   4
00228 INPUT_BAD:
                        .BLKB   4
0022C INPUT_BLOCK:
                        .BLKB   4
00230 INPUT_MAXBLOCK:
                        .BLKB   4
00234 INPUT_MEDIA_ID:
                        .BLKB   4
00238 INPUT_NAMEDESC:
                        .BLKB   8
00240 INPUT_STATBLK:
                        .BLKB   8
00248 INPUT_HDR_BEG:
                        .BLKB   0
00248 INPUT_CREDATE:
                        .BLKB   8
00250 INPUT_REVDATE:
                        .BLKB   8
00258 INPUT_EXPDATE:
                        .BLKB   8
00260 INPUT_BAKDATE:
                        .BLKB   8
```

```
00268 INPUT_FILEOWNER:
                .BLKB   4
0026C INPUT_FILECHAR:
                .BLKB   4
00270 INPUT_RECATTR:
                .BLKB   32
00290 INPUT_HDR_END:
                .BLKB   0
00290 INPUT_END:
                .BLKB   0
00290 INPUT_PROC_LIST:
                .BLKB   4
00294 INPUT_PLACEMENT:
                .BLKB   8
0029C INPUT_VBN_LIST:
                .BLKB   8
002A4 INPUT_PLACE_LEN:
                .BLKB   2
002A6 INPUT_PADDING_2:
                .BLKB   2
002A8 OUTPUT_BEG:
                .BLKB   0
002A8 OUTPUT_CHAN:
                .BLKB   4
002AC OUTPUT_FLAGS:
                .BLKB   2
002AE OUTPUT_PADDING:
                .BLKB   2
002B0 OUTPUT_FAB:
                .BLKB   4
002B4 OUTPUT_NAM:
                .BLKB   4
002B8 OUTPUT_BCB:
                .BLKB   4
002BC OUTPUT_QUAL:
                .BLKB   4
002C0 OUTPUT_BAD:
                .BLKB   4
002C4 OUTPUT_BLOCK:
                .BLKB   4
002C8 OUTPUT_MAXBLOCK:
                .BLKB   4
002CC OUTPUT_DEVGEOM:
                .BLKB   8
002D4 OUTPUT_ATTBUF:
                .BLKB   144
00364 OUTPUT_END:
                .BLKB   0
00364 LIST_TOTFILES:
                .BLKB   4
00368 LIST_TOTSIZE:
                .BLKB   4
0036C VERIFY_FAB:
                .BLKB   4
00370 VERIFY_USE_COUNT:
                .BLKB   4
00374 VERIFY_QUAL:
```

```
                        .BLKB   4
00378 COMPARE_BCB:
                        .BLKB   4
0037C FAST_BUFFER:
                        .BLKB   4
00380 FAST_BUFFER_SIZE:
                        .BLKB   4
00384 FAST_RVN:
                        .BLKB   1
00385 FAST_PADDING:
                        .BLKB   1
00386 DIR_VERLIMIT:
                        .BLKB   2
00388 FAST_VOL_BEG:
                        .BLKB   0
00388 FAST_IMAP_SIZE:
                        .BLKB   4
0038C FAST_IMAP:
                        .BLKB   4
00390 FAST_HDR_OFFSET:
                        .BLKB   4
00394 FAST_BOOT_LBN:
                        .BLKB   4
00398 FAST_VOL_END:
                        .BLKB   0
00398 JOUR_BUFFER:
                        .BLKB   4
0039C JOUR_DIR:
                        .BLKB   4
003A0 JOUR_HIBLK:
                        .BLKB   4
003A4 JOUR_EFBLK:
                        .BLKB   4
003A8 JOUR_INBLK:
                        .BLKB   4
003AC JOUR_FFBYTE:
                        .BLKB   2
003AE JOUR_INBYTE:
                        .BLKB   2
003B0 JOUR_STRUCT_LEV:
                        .BLKB   2
003B2 JOUR_COUNT:
                        .BLKB   1
003B3 JOUR_REVERSE:
                        .BLKB   1
003B4 JOUR_EXSZ:
                        .BLKB   2
003B6 JOUR_PADDING:
                        .BLKB   2
003B8 CHKPT_HIGH_SP:
                        .BLKB   4
003BC CHKPT_LOW_SP:
                        .BLKB   4
003C0 CHKPT_STACK:
                        .BLKB   4
003C4 CHKPT_VARS:
                        .BLKB   4
```

```
003C8 CHKPT_STATUS:
                .BLKB   4
003CC DIR_BEG:.BLKB     0
003CC DIR_CHAN:
                .BLKB   4
003D0 DIR_NAM:.BLKB     4
003D4 DIR_DEV_DESC:
                .BLKB   4
003D8 DIR_SEL_DIR:
                .BLKB   8
003E0 DIR_SEL_NTV:
                .BLKB   8
003E8 DIR_STRUCLEV:
                .BLKB   1
003E9 DIR_LEVELS:
                .BLKB   1
003EA D R_FLAGS:
                .BLKB   1
003EB DIR_STATUS:
                .BLKB   1
003EC DIR_STRING:
                .BLKB   320
0052C DIR_STACK:
                .BLKB   612
00790 DIR_SP: .BLKB
00794 DIR_SEL_LATEST:
                .BLKB   4
00798 DIR_END:.BLKB     0
00798 DIR_SCANLIMIT:
                .BLKB   36
007BC INPUT_MTL:
                .BLKB   4
007C0 OUTPUT_MTL:
                .BLKB   4
007C4 CURRENT_MTL:
                .BLKB   4
007C8 CURRENT_VCB:
                .BLKB   4
007CC CURRENT_WCB:
                .BLKB   4
007D0 ACL_FIB_DESCR:
                .BLKB   8
007D8 ACL_FIB:.BLKB     64
00818 ACL_LENGTH:
                .BLKB   4
0081C ACL_BUFFER:
                .BLKB   4
00820 CRYP_IN_CONTEXT:
                .BLKB   4
00824 CRYP_OU_CONTEXT:
                .BLKB   4
00828 CRYP_DA_CONTEXT:
                .BLKB   4
0082C CRYP_DATA_ENCIV:
                .BLKB   8
C0834 CRYP_DATA_CODE:
                .BLKB   4
```

```
                        00838 CRYP_DATA_KEY:
                                .BLKB   8
                        00840 CRYP_DATA_IV:
                                .BLKB   8
                        00848 CRYP_DATA_CKSM:
                                .BLKB   4

                                .PSECT  DATA,NOEXE,2

                        00000 RSA_DESC:
                                .BLKB   8
                        00008 DQF_QUOTA_FID:
                                .BLKB   6
                        0000E           .BLKB   2
                        00010 DQF_ROOT:
                                .BLKB   4
                        00014 DQF_COUNT:
                                .BLKB   4
                        00018 QUEUE_HEADERS:
                                .BLKB   24

                                .EXTRN  CHECKSUM, CHECKSUM2
                                .EXTRN  FILE_ERROR, FROM_ODS1_DATE
                                .EXTRN  INITIALIZE_VOLUME
                                .EXTRN  MAKE_NAMEBLOCK, FREE_VM
                                .EXTRN  GET_VM, GET_ZERO_VM
                                .EXTRN  SYS$ASSIGN, ACL_DELETEACL
                                .EXTRN  ACL_BUILDACL, ACL_DISPATCH
                                .EXTRN  BACKUP$_BADSETCNT
                                .EXTRN  BACKUP$_DISKFRAG
                                .EXTRN  BACKUP$_GETCHN, BACKUP$_INCRVN
                                .EXTRN  BACKUP$_INCSETCNT
                                .EXTRN  BACKUP$_INVATTVAL
                                .EXTRN  BACKUP$_INVFID, BACKUP$_INVHOMBLK
                                .EXTRN  BACKUP$_LARGECNT
                                .EXTRN  BACKUP$_NOBITMAP
                                .EXTRN  BACKUP$_NOHOMEBLK
                                .EXTRN  BACKUP$_NOINDEXF
                                .EXTRN  BACKUP$_NOVOLDATA
                                .EXTRN  BACKUP$_ODS2SAVE
                                .EXTRN  BACKUP$_OPENIN, BACKUP$_OPENOUT
                                .EXTRN  BACKUP$_CLOSEOUT
                                .EXTRN  BACKUP$_READBMAP
                                .EXTRN  BACKUP$_READERR
                                .EXTRN  BACKUP$_READIMAP
                                .EXTRN  BACKUP$_READYREAD
                                .EXTRN  BACKUP$_READYWRITE
                                .EXTRN  BACKUP$_STRUCLEV
                                .EXTRN  BACKUP$_VOLINSET
                                .EXTRN  BACKUP$_WRITENABLE
                                .EXTRN  BACKUP$_WRITEERR
                                .EXTRN  BACKUP$_QUOTAFILE

                                .PSECT  CODE,NOWRT,2

                        000C 00000 DQF_FIND_UIC:
                                .WORD   Save R2,R3                              ; 1876
```

```
                                53 00000000'    EF  9E 00002             MOVAB   DQF_ROOT, Q                        ; 1913
                                52              63  D0 00009 1$:         MOVL    (Q), P                            ; 1914
                                                14  13 0000C             BEQL    3$                                ; 1915
                          08    A2        04    AC  D1 0000E             CMPL    UIC, 8(P)                         ; 1917
                                                34  13 00013             BEQL    4$                                ;
                                                06  1B 00015             BLEQU   2$                                ; 1919
                                53        04    A2  9E 00017             MOVAB   4(R2), Q                          ; 1920
                                                EC  11 0001B             BRB     1$                                ;
                                53              52  D0 0001D 2$:         MOVL    P, Q                              ; 1921
                                                E7  11 00020             BRB     1$                                ; 1922
                                   00000000'    EF  D6 00022 3$:         INCL    DQF_COUNT                         ; 1928
                                                18  DD 00028             PUSHL   #24                               ; 1929
                  00000000G     00                01  FB 0002A             CALLS   #1, GET_VM                        ;
                                63                50  D0 00031             MOVL    R0, (Q)                           ;
                                52                63  D0 00034             MOVL    (Q), P                            ; 1930
                                                  62  7C 00037             CLRQ    (P)                               ; 1931
                          08    A2        04    AC  D0 00039             MOVL    UIC, 8(P)                         ; 1933
                                          0C    A2  D4 0003E             CLRL    12(P)                             ; 1934
                          10    A2                01  CE 00041             MNEGL   #1, 16(P)                         ; 1935
                          14    A2                01  CE 00045             MNEGL   #1, 20(P)                         ; 1936
                                50                52  D0 00049 4$:         MOVL    P, R0                             ; 1938
                                                  04  0004C               RET                                       ;

; Routine Size:  77 bytes,    Routine Base:  CODE + 0000
```

```
;  388          1939  1 %SBTTL 'DQF_MODIFY_USAGE - record disk space usage'
;  389          1940  1 ROUTINE DQF_MODIFY_USAGE (UIC,USAGE): NOVALUE=
;  390          1941  1
;  391          1942  1 !++
;  392          1943  1 !
;  393          1944  1 ! FUNCTIONAL DESCRIPTION:
;  394          1945  1 !       This routine records used blocks.
;  395          1946  1 !
;  396          1947  1 ! INPUT PARAMETERS:
;  397          1948  1 !       UIC             - The UIC.
;  398          1949  1 !       USAGE           - Count of blocks allocated.
;  399          1950  1 !
;  400          1951  1 ! IMPLICIT INPUTS:
;  401          1952  1 !       Disk quota table.
;  402          1953  1 !
;  403          1954  1 ! OUTPUT PARAMETERS:
;  404          1955  1 !       NONE
;  405          1956  1 !
;  406          1957  1 ! IMPLICIT OUTPUTS:
;  407          1958  1 !       Disk quota table updated.
;  408          1959  1 !
;  409          1960  1 ! ROUTINE VALUE:
;  410          1961  1 !       NONE
;  411          1962  1 !
;  412          1963  1 ! SIDE EFFECTS:
;  413          1964  1 !       NONE
;  414          1965  1 !
;  415          1966  1 !--
;  416          1967  1
;  417          1968  2 BEGIN
;  418          1969  2 LOCAL
;  419          1970  2         P:              REF BBLOCK;     ! Pointer to DQF entry
;  420          1971  2
;  421          1972  2
;  422          1973  2 P = DQF_FIND_UIC(.UIC);
;  423          1974  2 P[DQF_USAGE] = .P[DQF_USAGE] + .USAGE;
;  424          1975  1 END;


                                        0000 00000 DQF_MODIFY_USAGE:
                                                   .WORD   Save nothing                    ; 1940
                                    04  AC  DD 00002       PUSHL   UIC                     ; 1973
                                AA  AF      01  FB 00005   CALLS   #1, DQF_FIND_UIC        ;
                                0C  A0  08  AC  C0 00009   ADDL2   USAGE, T2(P)            ; 1974
                                            04 0000E       RET                             ; 1975

; Routine Size:  15 bytes,    Routine Base:  CODE + 004D
```

```
;  426          1976  1 %SBTTL 'DQF_WRITE_ENTRY - write out quota file entry'
;  427          1977  1 ROUTINE DQF_WRITE_ENTRY (P): L_DQF_WRITE_ENTRY NOVALUE=
;  428          1978  1
;  429          1979  1 !++
;  430          1980  1 !
;  431          1981  1 ! FUNCTIONAL DESCRIPTION:
;  432          1982  1 !       This routine writes a quota file entry.
;  433          1983  1 !
;  434          1984  1 ! INPUT PARAMETERS:
;  435          1985  1 !       P                   - Pointer to DQF entry.
;  436          1986  1 !
;  437          1987  1 ! IMPLICIT INPUTS:
;  438          1988  1 !       DQF_BUFFER          - Pointer to block buffer.
;  439          1989  1 !       DQF_RECORD          - Pointer to next available record.
;  440          1990  1 !       DQF_VBN             - Next VBN.
;  441          1991  1 !       DQF_DEFAULT_PERM    - Default PERMQUOTA.
;  442          1992  1 !       DQF_DEFAULT_OVER    - Default OVERDRAFT.
;  443          1993  1 !
;  444          1994  1 ! OUTPUT PARAMETERS:
;  445          1995  1 !       NONE
;  446          1996  1 !
;  447          1997  1 ! IMPLICIT OUTPUTS:
;  448          1998  1 !       NONE
;  449          1999  1 !
;  450          2000  1 ! ROUTINE VALUE:
;  451          2001  1 !       NONE
;  452          2002  1 !
;  453          2003  1 ! SIDE EFFECTS:
;  454          2004  1 !       NONE
;  455          2005  1 !
;  456          2006  1 !--
;  457          2007  1
;  458          2008  2 BEGIN
;  459          2009  2 MAP
;  460          2010  2         P:              REF BBLOCK;     ! Pointer to DQF entry
;  461          2011  2 LOCAL
;  462          2012  2         STATUS,                         ! General status variable
;  463          2013  2         IOSB:           VECTOR[4,WORD]; ! I/O status block
;  464          2014  2 EXTERNAL REGISTER
;  465          2015  2         DQF_BUFFER,
;  466          2016  2         DQF_RECORD:     REF BBLOCK,
;  467          2017  2         DQF_VBN,
;  468          2018  2         DQF_DEFAULT_PERM,
;  469          2019  2         DQF_DEFAULT_OVER;
;  470          2020  2
;  471          2021  2
;  472          2022  2 ! Recursively write all entries on the left branch.
;  473          2023  2 !
;  474          2024  2 IF .P[DQF_LLINK] NEQ 0 THEN DQF_WRITE_ENTRY(.P[DQF_LLINK]);
;  475          2025  2
;  476          2026  2
;  477          2027  2 IF .DQF_RECORD GEQA .DQF_BUFFER + 512
;  478          2028  2 THEN
;  479          2029  3     BEGIN
;  480        P 2030  3     STATUS = $$QIOW(
;  481        P 2031  3         FUNC=IO$_WRITEVBLK,
;  482        P 2032  3         CHAN=STA_OUT_CHAN,
```

```
;  483        P 2033  3          IOSB=IOSB,
;  484        P 2034  3          P1=.DQF_BUFFER,
;  485        P 2035  3          P2=512,
;  486          2036  3          P3=.DQF_VBN);
;  487          2037  3      IF .STATUS THEN STATUS = .IOSB[0];
;  488          2038  3      IF NOT .STATUS
;  489          2039  3      THEN
;  490          2040  3          SIGNAL(BACKUP$_WRITEERR + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;  491          2041  3      CH$FILL(0, 512, .DQF_BUFFER);
;  492          2042  3      DQF_RECORD = .DQF_BUFFER;
;  493          2043  3      DQF_VBN = .DQF_VBN + 1;
;  494          2044  2      END;
;  495          2045  2
;  496          2046  2
;  497          2047  2 DQF_RECORD[DQF$L_FLAGS] = DQF$M_ACTIVE;
;  498          2048  2 DQF_RECORD[DQF$L_UIC] = .P[DQF_UIC];
;  499          2049  2 DQF_RECORD[DQF$L_USAGE] = .P[DQF_USAGE];
;  500          2050  2 IF .P[DQF_PERMQUOTA] EQL -1 AND .P[DQF_OVERDRAFT] EQL -1
;  501          2051  2 THEN
;  502          2052  3     BEGIN
;  503          2053  3     DQF_RECORD[DQF$L_PERMQUOTA] = .DQF_DEFAULT_PERM;
;  504          2054  3     DQF_RECORD[DQF$L_OVERDRAFT] = .DQF_DEFAULT_OVER;
;  505          2055  3     END
;  506          2056  2 ELSE
;  507          2057  3     BEGIN
;  508          2058  3     DQF_RECORD[DQF$L_PERMQUOTA] = .P[DQF_PERMQUOTA];
;  509          2059  3     DQF_RECORD[DQF$L_OVERDRAFT] = .P[DQF_OVERDRAFT];
;  510          2060  2     END;
;  511          2061  2 DQF_RECORD = .DQF_RECORD + DQF$C_LENGTH;
;  512          2062  2
;  513          2063  2
;  514          2064  2 ! Recursively write all entries on the right branch.
;  515          2065  2 !
;  516          2066  2 IF .P[DQF_RLINK] NEQ 0 THEN DQF_WRITE_ENTRY(.P[DQF_RLINK]);
;  517          2067  1 END;


                                                .EXTRN  STA_QIOW

                                 007C 00000 DQF_WRITE_ENTRY:
                                                .WORD   Save R2,R3,R4,R5,R6             ; 1977
                              5E                C2 00002          SUBL2   #8, SP
                              56          04    AC D0 00005       MOVL    P, R6         ; 2024
                                                66 D5 00009       TSTL    (R6)
                                                06 13 0000B       BEQL    1$
                                                66 DD 0000D       PUSHL   (R6)
                           ED AF                01 FB 0000F       CALLS   #1, DQF_WRITE_ENTRY
                              50        0200    CB 9E 00013 1$:   MOVAB   512(R11), R0  ; 2027
                              50                5A D1 00018       CMPL    DQF_RECORD, R0
                                                50 1F 0001B       BLSSU   4$
                                                7E 7C 0001D       CLRQ    -(SP)         ; 2036
                                                7E D4 0001F       CLRL    -(SP)
                                                59 DD 00021       PUSHL   DQF_VBN
                              7E        0200    8F 3C 00023       MOVZWL  #512, -(SP)
                                                5B DD 00028       PUSHL   DQF_BUFFER
                                                7E 7C 0002A       CLRQ    -(SP)
```

```
                                        20   AE  9F 0002C          PUSHAB  IOSB                                                    ;
                                        30   DD  0002F          PUSHL   #48                                                     ;
                                  0002FFFF   8F  DD 00031          PUSHL   #196607                                                 ;
                                             7E  D4 00037          CLRL    -(SP)                                                   ;
                          00000000G     00   0C  FB 00039          CALLS   #12, STA_QIOW                                           ;
                                        06   50  E9 00040          BLBC    STATUS, 2$                                              ; 2037
                                        50   6E  3C 00043          MOVZWL  IOSB, STATUS                                            ;
                                        17   50  E8 00046          BLBS    STATUS, 3$                                              ; 2038
                                             50  DD 00049 2$:      PUSHL   STATUS                                                  ; 2040
                                  00000000'  EF  9F 0004B          PUSHAB  RSA_DESC                                                ;
                                             01  DD 00051          PUSHL   #1                                                      ;
                                  00000000G  8F  DD 00053          PUSHL   #BACKUP$_WRITEERR+2                                     ;
                          00000000G     00   04  FB 00059          CALLS   #4, LIB$SIGNAL                                          ;
    0200    8F          00            6E       00  2C 00060 3$:   MOVC5   #0, (SP), #0, #512, (DQF_BUFFER)                       ; 2041
                                             6B      00067                                                                       ;
                                        5A       5B  D0 00068          MOVL    DQF_BUFFER, DQF_RECORD                            ; 2042
                                                 59  D6 0006B          INCL    DQF_VBN                                           ; 2043
                                        6A       01  D0 0006D 4$:      MOVL    #1, (DQF_RECORD)                                  ; 2047
                                  04    AA   08  A6  7D 00070          MOVQ    8(R6), 4(DQF_RECORD)                              ; 2048
                              FFFFFFFF  8F   10  A6  D1 00075          CMPL    16(R6), #-1                                       ; 2050
                                                 14  12 0007D          BNEQ    5$                                                ;
                              FFFFFFFF  8F   14  A6  D1 0007F          CMPL    20(R6), #-1                                       ;
                                                 0A  12 00087          BNEQ    5$                                                ;
                                  0C    AA       58  D0 00089          MOVL    DQF_DEFAULT_PERM, 12(DQF_RECORD)                  ; 2053
                                  10    AA       57  D0 0008D          MOVL    DQF_DEFAULT_OVER, 16(DQF_RECORD)                  ; 2054
                                                 05  11 00091          BRB     6$                                                ; 2050
                                  0C    AA   10  A6  7D 00093 5$:      MOVQ    16(R6), 12(DQF_RECORD)                            ; 2058
                                        5A       20  C0 00098 6$:      ADDL2   #32, DQF_RECORD                                   ; 2061
                                             04  A6  D5 0009B          TSTL    4(R6)                                             ; 2066
                                                 08  13 0009E          BEQL    7$                                                ;
                                             04  A6  DD 000A0          PUSHL   4(R6)                                             ;
                                    FF58     CF       01  FB 000A3          CALLS   #1, DQF_WRITE_ENTRY                           ;
                                                      04 000A8 7$:      RET                                                       ; 2067

; Routine Size:  169 bytes,    Routine Base:  CODE + 005C
```

```
;  519          2068  1 %SBTTL 'ASSIGN_INPUT_CHANNEL - assign disk input channel'
;  520          2069  1 GLOBAL ROUTINE ASSIGN_INPUT_CHANNEL (DEVNAM,CHAN,ACMODE,MBXNAM)=
;  521          2070  1
;  522          2071  1 !++
;  523          2072  1 !
;  524          2073  1 ! FUNCTIONAL DESCRIPTION:
;  525          2074  1 !       This routine is called to assign a channel to the input disk.
;  526          2075  1 !
;  527          2076  1 ! INPUT PARAMETERS:
;  528          2077  1 !       As for the $ASSIGN system service.  (However, the channel number is
;  529          2078  1 !       written as a longword because the pseudo-channel numbers are larger
;  530          2079  1 !       than 16 bits).
;  531          2080  1 !
;  532          2081  1 ! IMPLICIT INPUTS:
;  533          2082  1 !       NONE
;  534          2083  1 !
;  535          2084  1 ! OUTPUT PARAMETERS:
;  536          2085  1 !       NONE
;  537          2086  1 !
;  538          2087  1 ! IMPLICIT OUTPUTS:
;  539          2088  1 !       NONE
;  540          2089  1 !
;  541          2090  1 ! ROUTINE VALUE:
;  542          2091  1 !       Completion status.
;  543          2092  1 !
;  544          2093  1 ! SIDE EFFECTS:
;  545          2094  1 !       NONE
;  546          2095  1 !
;  547          2096  1 !--
;  548          2097  1
;  549          2098  2 BEGIN
;  550          2099  2 BUILTIN
;  551          2100  2         AP;
;  552          2101  2
;  553          2102  2
;  554          2103  2 ! If the standalone ACP is going to handle operations on the input disk, then
;  555          2104  2 ! return the special channel number.  Otherwise, let the request through to
;  556          2105  2 ! the real $ASSIGN service.
;  557          2106  2 !
;  558          2107  2 IF .INPUT_MTL NEQ 0
;  559          2108  2 THEN
;  560          2109  3     BEGIN
;  561          2110  3     .CHAN = STA_IN_CHAN;
;  562          2111  3     SS$_NORMAL
;  563          2112  3     END
;  564          2113  2 ELSE
;  565          2114  2     CALLG(.AP, SYS$ASSIGN)
;  566          2115  1 END;


                                  0000 00000         .ENTRY  ASSIGN_INPUT_CHANNEL, Save nothing      ; 2069
                      00000000'   EF   D5 00002      TSTL    INPUT_MTL                               ; 2107
                                  0C   13 00008      BEQL    1$                                      ;
                08    BC 0001FFFF   8F   D0 0000A    MOVL    #131071, @CHAN                          ; 2110
```

```
                                50              01  D0 00012            MOVL    #1, R0                                  ; 2109
                                                    04 00015            RET                                             ;
                        00000000G  00           6C  FA 00016 1$:        CALLG   (AP), SYS$ASSIGN                        ; 2114
                                                    04 0001D            RET                                             ; 2115

; Routine Size:  30 bytes,    Routine Base:  CODE + 0105
```

```
;  568          2116  1 %SBTTL 'ASSIGN_OUTPUT_CHANNEL - assign disk output channel'
;  569          2117  1 GLOBAL ROUTINE ASSIGN_OUTPUT_CHANNEL (DEVNAM,CHAN,ACMODE,MBXNAM)=
;  570          2118  1
;  571          2119  1 !++
;  572          2120  1 !
;  573          2121  1 ! FUNCTIONAL DESCRIPTION:
;  574          2122  1 !       This routine is called to assign a channel to the output disk.
;  575          2123  1 !
;  576          2124  1 ! INPUT PARAMETERS:
;  577          2125  1 !       As for the $ASSIGN system service.  (However, the channel number is
;  578          2126  1 !       written as a longword because the pseudo-channel numbers are larger
;  579          2127  1 !       than 16 bits).
;  580          2128  1 !
;  581          2129  1 ! IMPLICIT INPUTS:
;  582          2130  1 !       NONE
;  583          2131  1 !
;  584          2132  1 ! OUTPUT PARAMETERS:
;  585          2133  1 !       NONE
;  586          2134  1 !
;  587          2135  1 ! IMPLICIT OUTPUTS:
;  588          2136  1 !       NONE
;  589          2137  1 !
;  590          2138  1 ! ROUTINE VALUE:
;  591          2139  1 !       Completion status.
;  592          2140  1 !
;  593          2141  1 ! SIDE EFFECTS:
;  594          2142  1 !       NONE
;  595          2143  1 !
;  596          2144  1 !--
;  597          2145  1
;  598          2146  2 BEGIN
;  599          2147  2 BUILTIN
;  600          2148  2         AP;
;  601          2149  2
;  602          2150  2
;  603          2151  2 ! If the standalone ACP is going to handle operations on the output disk, then
;  604          2152  2 ! return the special channel number.  Otherwise, let the request through to
;  605          2153  2 ! the real $ASSIGN service.
;  606          2154  2 !
;  607          2155  2 IF .OUTPUT_MTL NEQ 0
;  608          2156  2 THEN
;  609          2157  3     BEGIN
;  610          2158  3     .CHAN = STA_OUT_CHAN;
;  611          2159  3     SS$_NORMAL
;  612          2160  3     END
;  613          2161  2 ELSE
;  614          2162  2     CALLG(.AP, SYS$ASSIGN)
;  615          2163  1 END;


                                    0000 00000          .ENTRY  ASSIGN_OUTPUT_CHANNEL, Save nothing          ; 2117
                         00000000'  EF   D5 00002       TSTL    OUTPUT_MTL                                   ; 2155
                                    0C   13 00008       BEQL    1$                                           ;
                    08   BC 0002FFFF   8F   D0 0000A    MOVL    #196607, @CHAN                               ; 2158
```

```
                                50              01  D0 00012            MOVL    #1, R0                                  ; 2157
                                                    04 00015            RET                                             ;
                                                                                                                        ;
                      00000000G 00              6C  FA 00016 1$:        CALLG   (AP), SYS$ASSIGN                        ; 2162
                                                    04 0001D            RET                                             ; 2163

; Routine Size:  30 bytes,    Routine Base:  CODE + 0123
```

```
;  617          2164  1 %SBTTL 'SWITCH_VOLUME - switch to selected volume'
;  618          2165  1 GLOBAL ROUTINE SWITCH_VOLUME (RVN)=
;  619          2166  1
;  620          2167  1 !++
;  621          2168  1 !
;  622          2169  1 ! FUNCTIONAL DESCRIPTION:
;  623          2170  1 !       This routine switches to a specified relative volume.
;  624          2171  1 !
;  625          2172  1 ! INPUT PARAMETERS:
;  626          2173  1 !       RVN             - Relative volume number.
;  627          2174  1 !
;  628          2175  1 ! IMPLICIT INPUTS:
;  629          2176  1 !       CURRENT_MTL     - Pointer to MTL for selected volume set.
;  630          2177  1 !
;  631          2178  1 ! OUTPUT PARAMETERS:
;  632          2179  1 !       NONE
;  633          2180  1 !
;  634          2181  1 ! IMPLICIT OUTPUTS:
;  635          2182  1 !       NONE
;  636          2183  1 !
;  637          2184  1 ! ROUTINE VALUE:
;  638          2185  1 !       Channel number assigned to specified RVN.
;  639          2186  1 !
;  640          2187  1 ! SIDE EFFECTS:
;  641          2188  1 !       NONE
;  642          2189  1 !
;  643          2190  1 !--
;  644          2191  1
;  645          2192  2 BEGIN
;  646          2193  2 LOCAL
;  647          2194  2         STATUS,                         ! general status value
;  648          2195  2         XVCB:           REF BBLOCK,     ! VCB that will have channel deassigned
;  649          2196  2         VCB:            REF BBLOCK;     ! VCB being switched to
;  650          2197  2
;  651          2198  2
;  652          2199  2 ! Locate VCB being switched to.
;  653          2200  2 !
;  654          2201  2 CURRENT_VCB = VCB = .CURRENT_MTL[MTL_VCB(.RVN-.CURRENT_MTL[MTL_RVN_BASE])];
;  655          2202  2
;  656          2203  2
;  657          2204  2 IF .VCB[VCB_CHAN] NEQ 0
;  658          2205  2 THEN
;  659          2206  3     BEGIN
;  660          2207  3
;  661          2208  3     ! Volume already has channel assigned.  Make sure it is first (implemented
;  662          2209  3     ! as, not second) in the LRU list.
;  663          2210  3     !
;  664          2211  3     IF .CURRENT_MTL[MTL_CHAN_2] EQL .VCB
;  665          2212  3     THEN
;  666          2213  4         BEGIN
;  667          2214  4         CURRENT_MTL[MTL_CHAN_2] = .CURRENT_MTL[MTL_CHAN_1];
;  668          2215  4         CURRENT_MTL[MTL_CHAN_1] = .VCB;
;  669          2216  3         END;
;  670          2217  3     END
;  671          2218  2 ELSE
;  672          2219  3     BEGIN
;  673          2220  3
```

```
;  674          2221  3      ! No channel is assigned.  Take the channel away from the second entry
;  675          2222  3      ! in the LRU list. If it exists, wait for the I/O count to drop to zero
;  676          2223  3      ! and then deassign the channel.
;  677          2224  3      !
;  678          2225  3      XVCB = .CURRENT_MTL[MTL_CHAN_2];
;  679          2226  3      IF .XVCB NEQ 0
;  680          2227  3      THEN
;  681          2228  4          BEGIN
;  682          2229  4          WHILE TRUE DO
;  683          2230  5              BEGIN
;  684          2231  5              $CLREF(EFN=31);
;  685          2232  5              IF .XVCB[VCB_IOCOUNT] LEQ 0 THEN EXITLOOP;
;  686          2233  5              $WAITFR(EFN=31);
;  687          2234  4              END;
;  688          2235  4          $DASSGN(CHAN=.XVCB[VCB_CHAN]);
;  689          2236  4          XVCB[VCB_CHAN] = 0;
;  690          2237  3          END;
;  691          2238  3
;  692          2239  3      STATUS = $ASSIGN(DEVNAM=VCB[VCB_DEVICE], CHAN=VCB[VCB_CHAN]);
;  693          2240  3      IF NOT .STATUS
;  694          2241  3      THEN
;  695          2242  3          SIGNAL(
;  696          2243  4              (IF .VCB[VCB_OUTPUT]
;  697          2244  4                  THEN BACKUP$_OPENOUT + STS$K_SEVERE
;  698          2245  3                  ELSE BACKUP$_OPENIN + STS$K_SEVERE),
;  699          2246  3              1,
;  700          2247  3              VCB[VCB_DEVICE],
;  701          2248  3              .STATUS);
;  702          2249  3      CURRENT_MTL[MTL_CHAN_2] = .CURRENT_MTL[MTL_CHAN_1];
;  703          2250  3      CURRENT_MTL[MTL_CHAN_1] = .VCB;
;  704          2251  2      END;
;  705          2252  2
;  706          2253  2
;  707          2254  2 ! Return the channel number.
;  708          2255  2 !
;  709          2256  2 .VCB[VCB_CHAN]
;  710          2257  1 END;


                                                                .EXTRN  SYS$CLREF, SYS$WAITFR
                                                                .EXTRN  SYS$DASSGN

                                                001C 00000      .ENTRY  SWITCH_VOLUME, Save R2,R3,R4                    ; 2165
                                    54 00000000'  EF   9E 00002 MOVAB   CURRENT_MTL, R4
                                    50            64   D0 00009 MOVL    CURRENT_MTL, R0                             ; 2201
                                    51        30  A0   9A 0000C MOVZBL  48(R0), R1
                                51  51        04  AC   51   C3 00010 SUBL3   R1, RVN, R1
                                    53        34 A041  D0 00015 MOVL    52(R0)[R1], VCB
                                              04  A4   53   D0 0001A MOVL    VCB, CURRENT_VCB
                                                  08   A3   B5 0001E TSTW    8(VCB)                                  ; 2204
                                                  08   13   00021 BEQL    1$
                                    53        04   A0   D1 00023 CMPL    4(R0), VCB                              ; 2211
                                                  6B   12   00027 BNEQ    9$
                                                  62   11   00029 BRB     8$                                      ; 2214
                                    52        04   A0   D0 0002B 1$:    MOVL    4(R0), XVCB                     ; 2225
                                                  27   13   0002F BEQL    4$                                      ; 2226
```

```
                                                  1F  DD 00031 2$:     PUSHL   #31                                     ; 2231
                        00000000G  00             01  FB 00033         CALLS   #1, SYS$CLREF                           ;
                                             0A   A2  B5 0003A         TSTW    10(XVCB)                                ; 2232
                                                  0B  15 0003D         BLEQ    3$                                      ;
                                                  1F  DD 0003F         PUSHL   #31                                     ; 2233
                        00000000G  00             01  FB 00041         CALLS   #1, SYS$WAITFR                          ;
                                                  E7  11 00048         BRB     2$                                      ; 2229
                                   7E        08   A2  3C 0004A 3$:     MOVZWL  8(XVCB), -(SP)                          ; 2235
                        00000000G  00             01  FB 0004E         CALLS   #1, SYS$DASSGN                          ;
                                             08   A2  B4 00055         CLRW    8(XVCB)                                 ; 2236
                                                  7E  7C 00058 4$:     CLRQ    -(SP)                                   ; 2239
                                             08   A3  9F 0005A         PUSHAB  8(VCB)                                  ;
                                             20   A3  9F 0005D         PUSHAB  32(VCB)                                 ;
                        00000000G  00             04  FB 00060         CALLS   #4, SYS$ASSIGN                          ;
                                   20             50  E8 00067         BLBS    STATUS, 7$                              ; 2240
                                                  50  DD 0006A         PUSHL   STATUS                                  ; 2248
                                             20   A3  9F 0006C         PUSHAB  32(VCB)                                 ; 2247
                                                  01  DD 0006F         PUSHL   #1                                      ;
                                   08        07   A3  E9 00071         BLBC    7(VCB), 5$                              ; 2243
                                        00000000G   8F  DD 00075         PUSHL   #BACKUP$_OPENOUT+4                      ; 2244
                                                  06  11 0007B         BRB     6$                                      ;
                                        00000000G   8F  DD 0007D 5$:     PUSHL   #BACKUP$_OPENIN+4                       ; 2245
                        00000000G  00             04  FB 00083 6$:     CALLS   #4, LIB$SIGNAL                          ; 2247
                                   50             64  D0 0008A 7$:     MOVL    CURRENT_MTL, R0                         ; 2249
                              04   A0             60  D0 0008D 8$:     MOVL    (R0), 4(R0)                             ;
                                   60             53  D0 00091         MOVL    VCB, (R0)                               ; 2250
                                   50        08   A3  3C 00094 9$:     MOVZWL  8(VCB), R0                              ; 2257
                                                  04  00098         RET                                             ;

; Routine Size:  153 bytes,    Routine Base:  CODE + 0141
```

```
;  712          2258  1 %SBTTL 'VERIFY_HEADER - validate file header'
;  713          2259  1 ROUTINE VERIFY_HEADER (HEADER,FILE_ID)=
;  714          2260  1
;  715          2261  1 !++
;  716          2262  1 !
;  717          2263  1 ! FUNCTIONAL DESCRIPTION:
;  718          2264  1 !       This routine determines if the block given it is a valid file header.
;  719          2265  1 !
;  720          2266  1 ! INPUT PARAMETERS:
;  721          2267  1 !       HEADER          - Pointer to header.
;  722          2268  1 !       FILE_ID         - Purported file ID.
;  723          2269  1 !
;  724          2270  1 ! IMPLICIT INPUTS:
;  725          2271  1 !       CURRENT_MTL     - Pointer to MTL for selected volume set.
;  726          2272  1 !
;  727          2273  1 ! OUTPUT PARAMETERS:
;  728          2274  1 !       NONE
;  729          2275  1 !
;  730          2276  1 ! IMPLICIT OUTPUTS:
;  731          2277  1 !       NONE
;  732          2278  1 !
;  733          2279  1 ! ROUTINE VALUE:
;  734          2280  1 !       0 if invalid file header
;  735          2281  1 !       1 if valid file header
;  736          2282  1 !       2 if deleted file header
;  737          2283  1 !
;  738          2284  1 ! SIDE EFFECTS:
;  739          2285  1 !       NONE
;  740          2286  1 !
;  741          2287  1 !--
;  742          2288  1
;  743          2289  2 BEGIN
;  744          2290  2 MAP
;  745          2291  2         HEADER:         REF BBLOCK,     ! file header arg
;  746          2292  2         FILE_ID:        REF BBLOCK;     ! file ID arg
;  747          2293  2
;  748          2294  2
;  749          2295  2 ! First check the structure level.
;  750          2296  2 !
;  751          2297  2 IF .HEADER[FH2$B_STRUCLEV] NEQ .CURRENT_MTL[MTL_STRUCLEV]
;  752          2298  2 THEN
;  753          2299  2     RETURN 0;
;  754          2300  2
;  755          2301  2
;  756          2302  2 IF .CURRENT_MTL[MTL_STRUCLEV] EQL 2
;  757          2303  2 THEN
;  758          2304  3     BEGIN
;  759          2305  3
;  760          2306  3     ! Check the area offsets and the retrieval pointer use counts for
;  761          2307  3     ! consistency.
;  762          2308  3     !
;  763          2309  3     IF
;  764          2310  3         .HEADER[FH2$B_IDOFFSET] LSSU $BYTEOFFSET (FH2$L_HIGHWATER)/2 OR
;  765          2311  3         .HEADER[FH2$B_MPOFFSET] LSSU .HEADER[FH2$B_IDOFFSET] OR
;  766          2312  3         .HEADER[FH2$B_ACOFFSET] LSSU .HEADER[FH2$B_MPOFFSET] OR
;  767          2313  3         .HEADER[FH2$B_RSOFFSET] LSSU .HEADER[FH2$B_ACOFFSET] OR
;  768          2314  3         .HEADER[FH2$B_MAP_INUSE] GTRU .HEADER[FH2$B_ACOFFSET] - .HEADER[FH2$B_MPOFFSET]
```

```
;   769         2315  3      THEN
;   770         2316  3          RETURN 0;
;   771         2317  3
;   772         2318  3
;   773         2319  3      ! At this point, we have verified that the block at least once was a
;   774         2320  3      ! valid file header.
;   775         2321  3      !
;   776         2322  3      ! Look at the file number in the header. If zero, this is a
;   777         2323  3      ! deleted header.
;   778         2324  3      !
;   779         2325  3      IF
;   780         2326  3          .HEADER[FH2$W_FID_NUM] EQL 0 AND
;   781         2327  3          .HEADER[FH2$B_FID_NMX] EQL 0
;   782         2328  3      THEN
;   783         2329  3          RETURN 2;
;   784         2330  3
;   785         2331  3
;   786         2332  3      ! Now compute the header checksum.
;   787         2333  3      !
;   788         2334  3      IF NOT CHECKSUM(.HEADER)
;   789         2335  3      THEN
;   790         2336  3          RETURN 2;
;   791         2337  3
;   792         2338  3
;   793         2339  3      ! Check file number and file sequence number.
;   794         2340  3      !
;   795         2341  3      IF
;   796         2342  3          .HEADER[FH2$W_FID_NUM] NEQ .FILE_ID[FID$W_NUM] OR
;   797         2343  3          .HEADER[FH2$B_FID_NMX] NEQ .FILE_ID[FID$B_NMX] OR
;   798         2344  3          .HEADER[FH2$W_FID_SEQ] NEQ .FILE_ID[FID$W_SEQ]
;   799         2345  3      THEN
;   800         2346  3          RETURN 2;
;   801         2347  3      END
;   802         2348  2  ELSE
;   803         2349  3      BEGIN
;   804         2350  3      LOCAL
;   805         2351  3          MAP_AREA:       REF BBLOCK;
;   806         2352  3
;   807         2353  3
;   808         2354  3      ! Now point to the map area and make sure that the extension
;   809         2355  3      ! RVN is zero.  Also check the retrieval pointer format data.
;   810         2356  3      !
;   811         2357  3      MAP_AREA = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
;   812         2358  3      IF
;   813         2359  3          .MAP_AREA[FM1$B_EX_RVN] NEQ 0 OR
;   814         2360  3          .MAP_AREA[FM1$B_COUNTSIZE] NEQ 1 OR
;   815         2361  3          .MAP_AREA[FM1$B_LBNSIZE] NEQ 3
;   816         2362  3      THEN
;   817         2363  3          RETURN 0;
;   818         2364  3
;   819         2365  3
;   820         2366  3      ! Check the retrieval pointer counts for consistency with the
;   821         2367  3      ! available space.
;   822         2368  3      !
;   823         2369  3      IF
;   824         2370  3          .MAP_AREA[FM1$B_INUSE] GTRU .MAP_AREA[FM1$B_AVAIL] OR
;   825         2371  3          .MAP_AREA[FM1$B_AVAIL] GTRU 255 - (.MAP_AREA + FM1$C_POINTERS - .HEADER) / 2
```

```
;   826          2372  3     THEN
;   827          2373  3         RETURN 0;
;   828          2374  3
;   829          2375  3
;   830          2376  3     ! At this point, we have verified that the block at least once was a
;   831          2377  3     ! valid file header.
;   832          2378  3     !
;   833          2379  3     ! Look at the file number in the header. If zero, this is a
;   834          2380  3     ! deleted header.
;   835          2381  3     !
;   836          2382  3     IF .HEADER[FH1$W_FID_NUM] EQL 0
;   837          2383  3     THEN
;   838          2384  3         RETURN 2;
;   839          2385  3
;   840          2386  3
;   841          2387  3     ! Now compute the header checksum.
;   842          2388  3     !
;   843          2389  3     IF NOT CHECKSUM(.HEADER)
;   844          2390  3     THEN
;   845          2391  3         RETURN 2;
;   846          2392  3
;   847          2393  3
;   848          2394  3     ! Check file number and file sequence number.
;   849          2395  3     !
;   850          2396  3     IF
;   851          2397  3         .HEADER[FH1$W_FID_NUM] NEQ .FILE_ID[FID$W_NUM] OR
;   852          2398  3         .HEADER[FH1$W_FID_SEQ] NEQ .FILE_ID[FID$W_SEQ]
;   853          2399  3     THEN
;   854          2400  3         RETURN 2;
;   855          2401  2     END;
;   856          2402  2
;   857          2403  2
;   858          2404  2 ! Header is OK.
;   859          2405  2 !
;   860          2406  2 RETURN 1;
;   861          2407  1 END;


                                001C 00000 VERIFY_HEADER:
                                           .WORD   Save R2,R3,R4                    ; 2259
                54 00000000G    00   9E 00002  MOVAB   CHECKSUM, R4                ;
                52         04   AC   D0 00009  MOVL    HEADER, R2                  ; 2297
                50 00000000'    EF   D0 0000D  MOVL    CURRENT_MTL, R0             ;
             1E AO         07   A2   91 00014  CMPB    7(R2), 30(R0)               ;
                                76   12 00019  BNEQ    5$                          ;
                53         08   AC   D0 0001B  MOVL    FILE_ID, R3                 ; 2342
                02         1E   A0   91 0001F  CMPB    30(R0), #2                  ; 2302
                                55   12 00023  BNEQ    4$                          ;
                26              62   91 00025  CMPB    (R2), #38                   ; 2310
                                12   1F 00028  BLSSU   1$                          ;
                62         01   A2   91 0002A  CMPB    1(R2), (R2)                 ; 2311
                                0C   1F 0002E  BLSSU   1$                          ;
             01 A2         02   A2   91 00030  CMPB    2(R2), 1(R2)                ; 2312
                                05   1F 00035  BLSSU   1$                          ;
```

```
                     02   A2        03   A2  91 00037          CMPB    3(R2), 2(R2)                        ; 2313
                                         03  1E 0003C 1$:      BGEQU   2$                                  ;
                                    0092  31 0003E           BRW     9$                                  ;
                          51   02   A2   9A 00041 2$:      MOVZBL  2(R2), R1                           ; 2314
                          50   01   A2   9A 00045           MOVZBL  1(R2), R0                           ;
                          51        50   C2 00049           SUBL2   R0, R1                              ;
          51        3A    A2        08   00  ED 0004C           CMPZV   #0, #8, 58(R2), R1                  ;
                                         7F  1A 00052           BGTRU   9$                                  ;
                                    08   A2  B5 00054           TSTW    8(R2)                               ; 2326
                                         05  12 00057           BNEQ    3$                                  ;
                                    0D   A2  95 00059           TSTB    13(R2)                              ; 2327
                                         6D  13 0005C           BEQL    7$                                  ;
                                         52  DD 0005E 3$:       PUSHL   R2                                  ; 2334
                          64             01  FB 00060           CALLS   #1, CHECKSUM                        ;
                          65             50  E9 00063           BLBC    R0, 7$                              ;
                          63   08        A2  B1 00066           CMPW    8(R2), (R3)                         ; 2342
                                         5F  12 0006A           BNEQ    7$                                  ;
                     05   A3        0D   A2  91 0006C           CMPB    13(R2), 5(R3)                       ; 2343
                                         58  12 00071           BNEQ    7$                                  ;
                     02   A3        0A   A2  B1 00073           CMPW    10(R2), 2(R3)                       ; 2344
                                         4F  11 00078           BRB     6$                                  ;
                          50   01   A2   9A 0007A 4$:      MOVZBL  1(R2), R0                           ; 2357
                          50        6240  3E 0007E           MOVAW   (R2)[R0], MAP_AREA                  ;
                                    01   A0  95 00082           TSTB    1(MAP_AREA)                         ; 2359
                                         4C  12 00085           BNEQ    9$                                  ;
                          01        06   A0  91 00087           CMPB    6(MAP_AREA), #1                     ; 2360
                                         46  12 0008B           BNEQ    9$                                  ;
                          03        07   A0  91 0008D           CMPB    7(MAP_AREA), #3                     ; 2361
                                         40  12 00091 5$:       BNEQ    9$                                  ;
                     09   A0        08   A0  91 00093           CMPB    8(MAP_AREA), 9(MAP_AREA)            ; 2370
                                         39  1A 00098           BGTRU   9$                                  ;
               51         52             50  C3 0009A           SUBL3   MAP_AREA, R2, R1                    ; 2371
                          51             0A  C2 0009E           SUBL2   #10, R1                             ;
                          51             02  C6 000A1           DIVL2   #2, R1                              ;
                          51        00FF C1  9E 000A4           MOVAB   255(R1), R1                         ;
          51        09   A0             08   00  ED 000A9           CMPZV   #0, #8, 9(MAP_AREA), R1             ;
                                         22  1A 000AF           BGTRU   9$                                  ;
                                    02   A2  B5 000B1           TSTW    2(R2)                               ; 2382
                                         15  13 000B4           BEQL    7$                                  ;
                                         52  DD 000B6           PUSHL   R2                                  ; 2389
                          64             01  FB 000B8           CALLS   #1, CHECKSUM                        ;
                          0D             50  E9 000BB           BLBC    R0, 7$                              ;
                          63   02        A2  B1 000BE           CMPW    2(R2), (R3)                         ; 2397
                                         07  12 000C2           BNEQ    7$                                  ;
                     02   A3        04   A2  B1 000C4           CMPW    4(R2), 2(R3)                        ; 2398
                                         04  13 000C9 6$:       BEQL    8$                                  ;
                          50             02  D0 000CB 7$:       MOVL    #2, R0                              ; 2400
                                         04  000CE           RET                                         ;
                          50             01  D0 000CF 8$:       MOVL    #1, R0                              ; 2406
                                         04  000D2           RET                                         ;
                                         50  D4 000D3 9$:       CLRL    R0                                  ; 2407
                                         04  000D5           RET                                         ;

; Routine Size:  214 bytes,    Routine Base:  CODE + 01DA
```

```
;  863          2408  1 %SBTTL 'READ_HEADER - read file header'
;  864          2409  1 GLOBAL ROUTINE READ_HEADER (FILE_ID,BUFFER)=
;  865          2410  1
;  866          2411  1 !++
;  867          2412  1 !
;  868          2413  1 ! FUNCTIONAL DESCRIPTION:
;  869          2414  1 !       This routine reads one file header into the specified buffer.
;  870          2415  1 !
;  871          2416  1 ! INPUT PARAMETERS:
;  872          2417  1 !       FILE_ID         - File ID of header to be read.
;  873          2418  1 !       BUFFER          - Pointer to buffer.
;  874          2419  1 !
;  875          2420  1 ! IMPLICIT INPUTS:
;  876          2421  1 !       CURRENT_MTL     - Pointer to MTL for selected volume set.
;  877          2422  1 !
;  878          2423  1 ! OUTPUT PARAMETERS:
;  879          2424  1 !       NONE
;  880          2425  1 !
;  881          2426  1 ! IMPLICIT OUTPUTS:
;  882          2427  1 !       NONE
;  883          2428  1 !
;  884          2429  1 ! ROUTINE VALUE:
;  885          2430  1 !       True if header successfully read, false otherwise.
;  886          2431  1 !
;  887          2432  1 ! SIDE EFFECTS:
;  888          2433  1 !       Header read into buffer.
;  889          2434  1 !
;  890          2435  1 !--
;  891          2436  1
;  892          2437  2 BEGIN
;  893          2438  2 MAP
;  894          2439  2         FILE_ID:        REF BBLOCK,     ! Pointer to file ID
;  895          2440  2         BUFFER:         REF BBLOCK;     ! Pointer to header
;  896          2441  2 LOCAL
;  897          2442  2         STATUS,                         ! Status variable
;  898          2443  2         IOSB:           VECTOR[4,WORD], ! I/O status block
;  899          2444  2         FILE_NUMBER,                    ! Clean file number
;  900          2445  2         RVN,                            ! Clean RVN
;  901          2446  2         VCB:            REF BBLOCK;     ! VCB for specified RVN
;  902          2447  2
;  903          2448  2
;  904          2449  2 ! Get a clean file number and RVN and validity check.  If failure,
;  905          2450  2 ! return failure.
;  906          2451  2 !
;  907          2452  2 FILE_NUMBER = .FILE_ID[FID$W_NUM];
;  908          2453  2 FILE_NUMBER<16,8> = .FILE_ID[FID$B_NMX];
;  909          2454  2 RVN = .FILE_ID[FID$B_RVN];
;  910          2455  2 IF .RVN - .CURRENT_MTL[MTL_RVN_BASE] GEQU .CURRENT_MTL[MTL_SETCOUNT]
;  911          2456  2 THEN RETURN SS$_DEVNOTMOUNT;
;  912          2457  2 VCB = .CURRENT_MTL[MTL_VCB(.RVN-.CURRENT_MTL[MTL_RVN_BASE])];
;  913          2458  2 IF .FILE_NUMBER GTRU .VCB[VCB_MAXFILIDX] THEN RETURN SS$_NOSUCHFILE;
;  914          2459  2
;  915          2460  2
;  916          2461  2 ! Read the header.
;  917          2462  2 !
;  918          2463  2 CURRENT_WCB = .VCB[VCB_INDEXF];
;  919          2464  2 STATUS = R_W_VIRTUAL(
```

```
;  920          2465  2      0,
;  921          2466  2      0,
;  922          2467  2      IO$_READVBLK,
;  923          2468  2      IOSB,
;  924          2469  2      0,
;  925          2470  2      0,
;  926          2471  2      .BUFFER,
;  927          2472  2      512,
;  928          2473  2      .FILE_NUMBER + .VCB[VCB_HDR_OFFSET]);
;  929          2474  2 $WAITFR(EFN=0);
;  930          2475  2 IF .STATUS THEN STATUS = .IOSB[0];
;  931          2476  2
;  932          2477  2
;  933          2478  2 ! If failure, return failure.
;  934          2479  2 !
;  935          2480  2 IF NOT .STATUS
;  936          2481  2 THEN
;  937          2482  2     RETURN .STATUS;
;  938          2483  2
;  939          2484  2
;  940          2485  2 ! Verify the header that was read.
;  941          2486  2 !
;  942          2487  2 IF NOT VERIFY_HEADER(.BUFFER, .FILE_ID)
;  943          2488  2 THEN
;  944          2489  2     RETURN SS$_NOSUCHFILE;
;  945          2490  2
;  946          2491  2
;  947          2492  2 ! Successful completion.
;  948          2493  2 !
;  949          2494  2 SS$_NORMAL
;  950          2495  1 END;


                                                001C 00000          .ENTRY  READ_HEADER, Save R2,R3,R4                ; 2409
                                             5E    08   C2 00002    SUBL2   #8, SP                                    ;
                                             53    04   AC DO 00005 MOVL    FILE_ID, R3                               ; 2452
                                             52         63 3C 00009 MOVZWL  (R3), FILE_NUMBER                         ;
                                52           08   10    05   A3 F0 0000C INSV    5(R3), #16, #8, FILE_NUMBER       ; 2453
                                             50    04   A3 9A 00012 MOVZBL  4(R3), RVN                               ; 2454
                                             51 00000000' EF DO 00016 MOVL    CURRENT_MTL, R1                           ; 2455
                                             54    30   A1 9A 0001D MOVZBL  48(R1), R4                                ;
                                             50         54 C2 00021 SUBL2   R4, R0                                    ;
                                50      1F   A1    08         00 ED 00024 CMPZV   #0, #8, 31(R1), R0                   ;
                                                           05 1A 0002A BGTRU   1$                                      ;
                                             50    7C   8F 9A 0002C MOVZBL  #124, R0                                 ; 2456
                                                           04 00030 RET                                                ;
                                             50    34 A140 DO 00031 1$:  MOVL    52(R1)[R0], VCB                     ; 2457
                                      1C     A0         52 D1 00036       CMPL    FILE_NUMBER, 28(VCB)                ; 2458
                                                           4A 1A 0003A       BGTRU   4$                                  ;
                                00000000'    EF         60 DO 0003C       MOVL    (VCB), CURRENT_WCB                  ; 2463
                                             50    1A   A0 3C 00043       MOVZWL  26(VCB), R0                         ; 2473
                                                      6042 9F 00047       PUSHAB  (R0)[FILE_NUMBER]                   ;
                                             7E   0200  8F 3C 0004A       MOVZWL  #512, -(SP)                         ; 2464
                                                  08    AC DD 0004F       PUSHL   BUFFER                              ; 2471
```

```
                                 7E  7C 00052          CLRQ    -(SP)                                   ; 2464
                              14 AE  9F 00054          PUSHAB  IOSB                                    ;
                                 31  DD 00057          PUSHL   #49                                     ;
                                 7E  7C 00059          CLRQ    -(SP)                                   ;
                       0000V  CF     09  FB 0005B          CALLS   #9, R_W_VIRTUAL                     ;
                              52     50  D0 00060          MOVL    R0, STATUS                          ;
                                     7E  D4 00063          CLRL    -(SP)                               ; 2474
                 00000000G   00     01  FB 00065          CALLS   #1, SYS$WAITFR                      ;
                              06     52  E9 0006C          BLBC    STATUS, 2$                          ; 2475
                              52     6E  3C 0006F          MOVZWL  IOSB, STATUS                        ;
                              04     52  E8 00072          BLBS    STATUS, 3$                          ; 2480
                              50     52  D0 00075 2$:       MOVL    STATUS, R0                          ; 2482
                                     04  00078            RET                                         ;
                                 53  DD 00079 3$:       PUSHL   R3                                  ; 2487
                              08 AC  DD 0007B          PUSHL   BUFFER                                  ;
                       FEA7   CF     02  FB 0007E          CALLS   #2, VERIFY_HEADER                   ;
                              06     50  E8 00083          BLBS    R0, 5$                              ;
                              50 0910 8F  3C 00086 4$:       MOVZWL  #2320, R0                       ; 2489
                                     04  0008B            RET                                         ;
                              50     01  D0 0008C 5$:       MOVL    #1, R0                              ; 2495
                                     04  0008F            RET                                         ;

; Routine Size:  144 bytes,    Routine Base:  CODE + 02B0
```

```
;  952          2496  1 %SBTTL 'WRITE_HEADER - write file header'
;  953          2497  1 ROUTINE WRITE_HEADER (FILE_ID,BUFFER)=
;  954          2498  1
;  955          2499  1 !++
;  956          2500  1 !
;  957          2501  1 ! FUNCTIONAL DESCRIPTION:
;  958          2502  1 !       This routine writes one file header from the specified buffer.
;  959          2503  1 !
;  960          2504  1 ! INPUT PARAMETERS:
;  961          2505  1 !       FILE_ID         - File ID of header to be written.
;  962          2506  1 !       BUFFER          - Pointer to buffer.
;  963          2507  1 !
;  964          2508  1 ! IMPLICIT INPUTS:
;  965          2509  1 !       CURRENT_MTL     - Pointer to MTL for selected volume set.
;  966          2510  1 !
;  967          2511  1 ! OUTPUT PARAMETERS:
;  968          2512  1 !       NONE
;  969          2513  1 !
;  970          2514  1 ! IMPLICIT OUTPUTS:
;  971          2515  1 !       NONE
;  972          2516  1 !
;  973          2517  1 ! ROUTINE VALUE:
;  974          2518  1 !       Completion status.
;  975          2519  1 !
;  976          2520  1 ! SIDE EFFECTS:
;  977          2521  1 !       Header written from buffer.
;  978          2522  1 !
;  979          2523  1 !--
;  980          2524  1
;  981          2525  2 BEGIN
;  982          2526  2 MAP
;  983          2527  2         FILE_ID:        REF BBLOCK,     ! Pointer to file ID
;  984          2528  2         BUFFER:         REF BBLOCK;     ! Pointer to header
;  985          2529  2 LOCAL
;  986          2530  2         STATUS,                         ! Status variable
;  987          2531  2         IOSB:           VECTOR[4,WORD], ! I/O status block
;  988          2532  2         FILE_NUMBER,                    ! Clean file number
;  989          2533  2         RVN,                            ! Clean RVN
;  990          2534  2         VCB:            REF BBLOCK;     ! VCB for specified RVN
;  991          2535  2
;  992          2536  2
;  993          2537  2 ! Get a clean file number and RVN and validity check.  If failure,
;  994          2538  2 ! do nothing.
;  995          2539  2 !
;  996          2540  2 FILE_NUMBER = .FILE_ID[FID$W_NUM];
;  997          2541  2 FILE_NUMBER<16,8> = .FILE_ID[FID$B_NMX];
;  998          2542  2 RVN = .FILE_ID[FID$B_RVN];
;  999          2543  2 IF .RVN - .CURRENT_MTL[MTL_RVN_BASE] GEQU .CURRENT_MTL[MTL_SETCOUNT]
; 1000          2544  2 THEN RETURN SS$_DEVNOTMOUNT;
; 1001          2545  2 VCB = .CURRENT_MTL[MTL_VCB(.RVN-.CURRENT_MTL[MTL_RVN_BASE])];
; 1002          2546  2 IF .FILE_NUMBER GTRU .VCB[VCB_MAXFILIDX] THEN RETURN SS$_NOSUCHFILE;
; 1003          2547  2
; 1004          2548  2
; 1005          2549  2 ! Recompute the checksum.
; 1006          2550  2 !
; 1007          2551  2 CHECKSUM(.BUFFER);
; 1008          2552  2
```

```
;  1009          2553  2
;  1010          2554  2 ! Write the block.
;  1011          2555  2 !
;  1012          2556  2 CURRENT_WCB = .VCB[VCB_INDEXF];
;  1013          2557  2 STATUS = R_W_VIRTUAL(
;  1014          2558  2     0,
;  1015          2559  2     0,
;  1016          2560  2     IO$_WRITEVBLK,
;  1017          2561  2     IOSB,
;  1018          2562  2     0,
;  1019          2563  2     0,
;  1020          2564  2     .BUFFER,
;  1021          2565  2     512,
;  1022          2566  2     .FILE_NUMBER + .VCB[VCB_HDR_OFFSET]);
;  1023          2567  2 $WAITFR(EFN=0);
;  1024          2568  2 IF .STATUS THEN STATUS = .IOSB[0];
;  1025          2569  2
;  1026          2570  2
;  1027          2571  2 ! If failure, return failure.
;  1028          2572  2 !
;  1029          2573  2 IF NOT .STATUS
;  1030          2574  2 THEN
;  1031          2575  2     RETURN .STATUS;
;  1032          2576  2
;  1033          2577  2
;  1034          2578  2 ! Set the index file bitmap bit.
;  1035          2579  2 !
;  1036          2580  2 IF .VCB[VCB_IMAP] NEQ 0
;  1037          2581  2 THEN
;  1038          2582  2     IF
;  1039          2583  3         (IF .BUFFER[FH2$B_STRUCLEV] EQL 2
;  1040          2584  3             THEN .BUFFER[FH2$W_FID_NUM] EQL 0
;  1041          2585  3             ELSE .BUFFER[FH1$W_FID_NUM] EQL 0)
;  1042          2586  2     THEN
;  1043          2587  2         BITVECTOR[.VCB[VCB_IMAP], .FILE_NUMBER-1] = FALSE
;  1044          2588  2     ELSE
;  1045          2589  2         BITVECTOR[.VCB[VCB_IMAP], .FILE_NUMBER-1] = TRUE;
;  1046          2590  2
;  1047          2591  2
;  1048          2592  2 ! If this is the index file header, also rewrite
;  1049          2593  2 ! the alternate header.
;  1050          2594  2 !
;  1051          2595  2 IF .FILE_NUMBER EQL FID$C_INDEXF AND .VCB[VCB_ODS_2]
;  1052          2596  2 THEN
;  1053          2597  3     BEGIN
;  1054          2598  3     STATUS = R_W_VIRTUAL(
;  1055          2599  3         0,
;  1056          2600  3         0,
;  1057          2601  3         IO$_WRITEVBLK,
;  1058          2602  3         IOSB,
;  1059          2603  3         0,
;  1060          2604  3         0,
;  1061          2605  3         .BUFFER,
;  1062          2606  3         512,
;  1063          2607  3         .VCB[VCB_CLUSTER] * 3 + 1);
;  1064          2608  3     $WAITFR(EFN=0);
;  1065          2609  3     IF .STATUS THEN STATUS = .IOSB[0];
```

```
;  1066          2610  3
;  1067          2611  3
;  1068          2612  3      ! If failure, return failure.
;  1069          2613  3      !
;  1070          2614  3      IF NOT .STATUS
;  1071          2615  3      THEN
;  1072          2616  3          RETURN .STATUS;
;  1073          2617  2      END;
;  1074          2618  2
;  1075          2619  2
;  1076          2620  2 ! Normal completion.
;  1077          2621  2 !
;  1078          2622  2 SS$_NORMAL
;  1079          2623  1 END;


                                        007C 00000 WRITE_HEADER:
                                                   .WORD   Save R2,R3,R4,R5,R6                     ; 2497
                          56 00000000G   00  9E 00002      MOVAB   SYS$WAITFR, R6                  ;
                          5E             08  C2 00009      SUBL2   #8, SP                          ;
                          50       04    AC  D0 0000C      MOVL    FILE_ID, R0                     ; 2540
                          53             60  3C 00010      MOVZWL  (R0), FILE_NUMBER               ;
                53        08           10  05  A0  F0 00013  INSV    5(R0), #16, #8, FILE_NUMBER     ; 2541
                          50       04    A0  9A 00019      MOVZBL  4(R0), RVN                      ; 2542
                          51 00000000'   EF  D0 0001D      MOVL    CURRENT_MTL, R1                 ; 2543
                          52       30    A1  9A 00024      MOVZBL  48(R1), R2                      ;
                          50             52  C2 00028      SUBL2   R2, R0                          ;
                50        1F   A1        08  00  ED 0002B  CMPZV   #0, #8, 31(R1), R0             ;
                                           05  1A 00031      BGTRU   1$                              ;
                          50       7C    8F  9A 00033      MOVZBL  #124, R0                        ; 2544
                                           04  00037      RET                                     ;
                          52       34  A140  D0 00038  1$:  MOVL    52(R1)[R0], VCB                 ; 2545
                     1C   A2             53  D1 0003D      CMPL    FILE_NUMBER, 28(VCB)            ; 2546
                                           06  1B 00041      BLEQU   2$                              ;
                          50     0910    8F  3C 00043      MOVZWL  #2320, R0                       ;
                                           04  00048      RET                                     ;
                          54       08    AC  D0 00049  2$:  MOVL    BUFFER, R4                      ; 2551
                                           54  DD 0004D      PUSHL   R4                              ;
                00000000G  00              01  FB 0004F      CALLS   #1, CHECKSUM                    ;
                00000000'  EF              62  D0 00056      MOVL    (VCB), CURRENT_WCB              ; 2556
                          50       1A    A2  3C 0005D      MOVZWL  26(VCB), R0                     ; 2566
                                       6043  9F 00061      PUSHAB  (R0)[FILE_NUMBER]               ;
                          7E     0200    8F  3C 00064      MOVZWL  #512, -(SP)                     ; 2557
                                           54  DD 00069      PUSHL   R4                              ; 2564
                                           7E  7C 0006B      CLRQ    -(SP)                           ; 2557
                                       14  AE  9F 0006D      PUSHAB  IOSB                            ;
                                           30  DD 00070      PUSHL   #48                             ;
                                           7E  7C 00072      CLRQ    -(SP)                           ;
                     0000V  CF             09  FB 00074      CALLS   #9, R_W_VIRTUAL                 ;
                          55             50  D0 00079      MOVL    R0, STATUS                      ;
                                           7E  D4 0007C      CLRL    -(SP)                           ; 2567
                          66             01  FB 0007E      CALLS   #1, SYS$WAITFR                  ;
                          64             55  E9 00081      BLBC    STATUS, 7$                      ; 2568
                          55             6E  3C 00084      MOVZWL  IOSB, STATUS                    ;
```

```
                                5E          55  E9 00087         BLBC    STATUS, 7$                          ; 2573
                                50      10  A2  D0 0008A         MOVL    16(VCB), R0                         ; 2580
                                            1E  13 0008E         BEQL    6$                                  ;
                                51      FF  A3  9E 00090         MOVAB   -1(R3), R1                          ; 2587
                                02      07  A4  91 00094         CMPB    7(R4), #2                           ; 2583
                                            05  12 00098         BNEQ    3$                                  ;
                                        08  A4  B5 0009A         TSTW    8(R4)                               ; 2584
                                            03  11 0009D         BRB     4$                                  ;
                                        02  A4  B5 0009F 3$:     TSTW    2(R4)                               ; 2585
                                            06  12 000A2 4$:     BNEQ    5$                                  ;
                    06                  60      51  E5 000A4         BBCC    R1, (R0), 6$                        ; 2587
                                            04  11 000A8         BRB     6$                                  ;
                    00                  60      51  E2 000AA 5$:     BBSS    R1, (R0), 6$                        ; 2589
                                        01  53  D1 000AE 6$:     CMPL    FILE_NUMBER, #1                     ; 2595
                                            39  12 000B1         BNEQ    8$                                  ;
                    34          07      A2      01  E1 000B3         BBC     #1, 7(VCB), 8$                      ;
                                52      04  A2  3C 000B8         MOVZWL  4(VCB), R2                          ; 2607
                                52          03  C4 000BC         MULL2   #3, R2                              ;
                                        01  A2  9F 000BF         PUSHAB  1(R2)                               ;
                                7E      0200    8F  3C 000C2         MOVZWL  #512, -(SP)                         ; 2598
                                        54  DD 000C7         PUSHL   R4                                  ; 2605
                                        7E  7C 000C9         CLRQ    -(SP)                               ; 2598
                                        14  AE  9F 000CB         PUSHAB  IOSB                                ;
                                        30  DD 000CE         PUSHL   #48                                 ;
                                        7E  7C 000D0         CLRQ    -(SP)                               ;
                        0000V   CF          09  FB 000D2         CALLS   #9, R_W_VIRTUAL                     ;
                                55          50  D0 000D7         MOVL    R0, STATUS                          ;
                                            7E  D4 000DA         CLRL    -(SP)                               ; 2608
                                66          01  FB 000DC         CALLS   #1, SYS$WAITFR                      ;
                                06          55  E9 000DF         BLBC    STATUS, 7$                          ; 2609
                                55          6E  3C 000E2         MOVZWL  IOSB, STATUS                        ;
                                04          55  E8 000E5         BLBS    STATUS, 8$                          ; 2614
                                50          55  D0 000E8 7$:     MOVL    STATUS, R0                          ; 2616
                                                04 000EB         RET                                         ;
                                50          01  D0 000EC 8$:     MOVL    #1, R0                              ; 2623
                                                04 000EF         RET                                         ;

; Routine Size:  240 bytes,    Routine Base:  CODE + 0340
```

```
;1081    2624  1 %SBTTL 'CREATE_DELHDR - format deleted header'
;1082    2625  1 ROUTINE CREATE_DELHDR (FILE_ID,HEADER): NOVALUE=
;1083    2626  1
;1084    2627  1 !++
;1085    2628  1 !
;1086    2629  1 ! FUNCTIONAL DESCRIPTION:
;1087    2630  1 !       This routine generates a deleted file header.
;1088    2631  1 !
;1089    2632  1 ! INPUT PARAMETERS:
;1090    2633  1 !       FILE_ID         - File ID of the header.
;1091    2634  1 !       HEADER          - Pointer to header buffer.
;1092    2635  1 !
;1093    2636  1 ! IMPLICIT INPUTS:
;1094    2637  1 !       NONE
;1095    2638  1 !
;1096    2639  1 ! OUTPUT PARAMETERS:
;1097    2640  1 !       NONE
;1098    2641  1 !
;1099    2642  1 ! IMPLICIT OUTPUTS:
;1100    2643  1 !       Header buffer contains deleted header.
;1101    2644  1 !
;1102    2645  1 ! ROUTINE VALUE:
;1103    2646  1 !       NONE
;1104    2647  1 !
;1105    2648  1 ! SIDE EFFECTS:
;1106    2649  1 !       NONE
;1107    2650  1 !
;1108    2651  1 !--
;1109    2652  1
;1110    2653  2 BEGIN
;1111    2654  2 MAP
;1112    2655  2         FILE_ID:        REF BBLOCK,     ! Pointer to file ID
;1113    2656  2         HEADER:         REF BBLOCK;     ! Pointer to header buffer
;1114    2657  2
;1115    2658  2
;1116    2659  2 CH$FILL(0, 512, .HEADER);
;1117    2660  2 IF .CURRENT_MTL[MTL_STRUCLEV] EQL 2
;1118    2661  2 THEN
;1119    2662  3     BEGIN
;1120    2663  3     HEADER[FH2$B_IDOFFSET] = FH2$C_LENGTH / 2;
;1121    2664  3     HEADER[FH2$B_MPOFFSET] = (FH2$C_LENGTH + FI2$C_LENGTH) / 2;
;1122    2665  3     HEADER[FH2$B_ACOFFSET] = $BYTEOFFSET(FH2$W_CHECKSUM) / 2;
;1123    2666  3     HEADER[FH2$B_RSOFFSET] = $BYTEOFFSET(FH2$W_CHECKSUM) / 2;
;1124    2667  3     HEADER[FH2$B_STRUCVER] = 1;
;1125    2668  3     HEADER[FH2$B_STRUCLEV] = 2;
;1126    2669  3     HEADER[FH2$W_FID_SEQ] = .FILE_ID[FID$W_SEQ];
;1127    2670  3     CH$COPY(
;1128    2671  3         2, UPLIT BYTE ('.;'),
;1129    2672  3         %C' ',
;1130    2673  3         FI2$S_FILENAME, BBLOCK[.HEADER + FH2$C_LENGTH, FI2$T_FILENAME]);
;1131    2674  3     CH$FILL (' ', FI2$S_FILENAMEXT,
;1132    2675  3             BBLOCK[.HEADER + FH2$C_LENGTH, FI2$T_FILENAMEXT]);
;1133    2676  3     END
;1134    2677  2 ELSE
;1135    2678  3     BEGIN
;1136    2679  3     HEADER[FH1$B_IDOFFSET] = FH1$C_LENGTH / 2;
;1137    2680  3     HEADER[FH1$B_MPOFFSET] = (FH1$C_LENGTH + FI1$C_LENGTH) / 2;
```

```
;  1138          2681  3     HEADER[FH1$W_FID_SEQ] = .FILE_ID[FID$W_SEQ];
;  1139          2682  3     HEADER[FH1$W_STRUCLEV] = FH1$C_LEVEL1;
;  1140          2683  3     BBLOCK[.HEADER + FH1$C_LENGTH * FI1$C_LENGTH, FM1$B_COUNTSIZE] = 1;
;  1141          2684  3     BBLOCK[.HEADER + FH1$C_LENGTH + FI1$C_LENGTH, FM1$B_LBNSIZE] = 3;
;  1142          2685  3     BBLOCK[.HEADER + FH1$C_LENGTH + FI1$C_LENGTH, FM1$B_AVAIL] =
;  1143          2686  3         (512-2-FH1$C_LENGTH-FI1$C_LENGTH-FM1$C_LENGTH)/2;
;  1144          2687  2     END;
;  1145          2688  2 CHECKSUM(.HEADER);
;  1146          2689  1 END;


                                                3B  2E  00430 P.AAA:  .ASCII  \.;\                                  ;


                                                    007C 00000 CREATE_DELHDR:
                                                                       .WORD   Save R2,R3,R4,R5,R6                     ; 2625
                                        56       08     AC  D0 00002       MOVL    HEADER, R6                      ; 2659
    0200  8F          00                6E             00  2C 00006       MOVC5   #0, (SP), #0, #512, (R6)        ;
                                                       66     0000D                                                ;
                                        51       04     AC  D0 0000E       MOVL    FILE_ID, R1                     ; 2669
                                        50 00000000'    EF  D0 00012       MOVL    CURRENT_MTL, R0                 ; 2660
                                        02       1E     A0  91 00019       CMPB    30(R0), #2                      ;
                                                        26  12 0001D       BNEQ    1$                              ;
                                        66 FFFF6428     8F  D0 0001F       MOVL    #-39896, (R6)                   ; 2663
                                  06    A6       0201   8F  B0 00026       MOVW    #513, 6(R6)                     ; 2667
                                  0A    A6         02   A1  B0 0002C       MOVW    2(R1), 10(R6)                   ; 2669
          14          20        C9    AF               02  2C 00031       MOVC5   #2, P.AAA, #32, #20, 80(R6)     ; 2673
                                                       A6     00037                                                ;
    0042  8F          20                6E             00  2C 00039       MOVC5   #0, (SP), #32, #66, 134(R6)     ; 2675
                                                 0086  C6     00040                                                ;
                                                       1A  11 00043       BRB     2$                              ; 2660
                                        66       2E17   8F  B0 00045 1$:    MOVW    #11799, (R6)                    ; 2679
                                  04    A6         02   A1  B0 0004A       MOVW    2(R1), 4(R6)                    ; 2681
                                  06    A6       0101   8F  B0 0004F       MOVW    #257, 6(R6)                     ; 2682
                                  62    A6       0301   8F  B0 00055       MOVW    #769, 98(R6)                    ; 2683
                                  65    A6              34  8E 0005B       MNEGB   #52, 101(R6)                    ; 2686
                                                        56  DD 0005F 2$:    PUSHL   R6                              ; 2688
                            00000000G   00              01  FB 00061       CALLS   #1, CHECKSUM                    ;
                                                            04 00068       RET                                     ; 2689

; Routine Size:  105 bytes,    Routine Base:  CODE + 0432
```

```
;1148           2690  1 %SBTTL 'TAKE_BLOCKS - remove blocks from free list'
;1149           2691  1 ROUTINE TAKE_BLOCKS (ACB,COUNT,LBN): NOVALUE=
;1150           2692  1
;1151           2693  1 !++
;1152           2694  1 !
;1153           2695  1 ! FUNCTIONAL DESCRIPTION:
;1154           2696  1 !       This routine modifies the free blocks list.
;1155           2697  1 !
;1156           2698  1 ! INPUT PARAMETERS:
;1157           2699  1 !       ACB             - Pointer to allocation control block.
;1158           2700  1 !       COUNT           - Count of blocks to allocate.
;1159           2701  1 !       LBN             - Logical block number of blocks to allocate.
;1160           2702  1 !
;1161           2703  1 ! IMPLICIT INPUTS:
;1162           2704  1 !       NONE
;1163           2705  1 !
;1164           2706  1 ! OUTPUT PARAMETERS:
;1165           2707  1 !       NONE
;1166           2708  1 !
;1167           2709  1 ! IMPLICIT OUTPUTS:
;1168           2710  1 !       NONE
;1169           2711  1 !
;1170           2712  1 ! ROUTINE VALUE:
;1171           2713  1 !       NONE
;1172           2714  1 !
;1173           2715  1 ! SIDE EFFECTS:
;1174           2716  1 !       Allocation list modified.
;1175           2717  1 !
;1176           2718  1 !--
;1177           2719  1
;1178           2720  2 BEGIN
;1179           2721  2 MAP
;1180           2722  2         ACB:            REF BBLOCK;             ! Pointer to allocation block
;1181           2723  2
;1182           2724  2
;1183           2725  2 IF .LBN EQL .ACB[ACB_LBN]
;1184           2726  2 THEN
;1185           2727  3     BEGIN
;1186           2728  3     !
;1187           2729  3     ! Allocation from beginning of free extent.
;1188           2730  3     ! If the entire extent is allocated, free the block.
;1189           2731  3     !
;1190           2732  3     ACB[ACB_LBN] = .ACB[ACB_LBN] + .COUNT;
;1191           2733  3     ACB[ACB_COUNT] = .ACB[ACB_COUNT] - .COUNT;
;1192           2734  3     IF .ACB[ACB_COUNT] EQL 0
;1193           2735  3     THEN
;1194           2736  4         BEGIN
;1195           2737  4         LOCAL
;1196           2738  4             DUMMY;                      ! Output for REMQUE
;1197           2739  4
;1198           2740  4         REMQUE(.ACB, DUMMY);
;1199           2741  4         FREE_VM(ACB_S_ENTRY, .ACB);
;1200           2742  3         END;
;1201           2743  3     END
;1202           2744  2 ELSE IF .ACB[ACB_LBN] + .ACB[ACB_COUNT] - .COUNT EQL .LBN
;1203           2745  2 THEN
;1204           2746  3     BEGIN
```

```
;  1205          2747  3      !
;  1206          2748  3      ! Allocation from end of free extent.
;  1207          2749  3      !
;  1208          2750  3      ACB[ACB_COUNT] = .ACB[ACB_COUNT] - .COUNT;
;  1209          2751  3      END
;  1210          2752  2  ELSE
;  1211          2753  3      BEGIN
;  1212          2754  3      !
;  1213          2755  3      ! Allocation from middle of free extent.
;  1214          2756  3      ! Generate a new block.
;  1215          2757  3      !
;  1216          2758  3      LOCAL
;  1217          2759  3          NEW:    REF BBLOCK;     ! Pointer to new ACB
;  1218          2760  3
;  1219          2761  3      NEW = GET_VM(ACB_S_ENTRY);
;  1220          2762  3      INSQUE(.NEW, .ACB);
;  1221          2763  3      NEW[ACB_LBN] = .LBN + .COUNT;
;  1222          2764  3      NEW[ACB_COUNT] = .ACB[ACB_LBN] + .ACB[ACB_COUNT] - .NEW[ACB_LBN];
;  1223          2765  3      ACB[ACB_COUNT] = .LBN - .ACB[ACB_LBN];
;  1224          2766  2      END;
;  1225          2767  1  END;
```

```
                                0004 00000 TAKE_BLOCKS:
                                                .WORD   Save R2                                         ; 2691
                        50      04    AC  D0 00002          MOVL    ACB, R0                           ; 2733
                        51      08    A0  9E 00006          MOVAB   8(R0), R1                         ;
                        50      04    AC  D0 0000A          MOVL    ACB, R0                           ; 2725
                    0C  A0      0C    AC  D1 0000E          CMPL    LBN, 12(R0)                       ;
                                      1C  12 00013          BNEQ    1$                                ;
                    0C  A0      08    AC  C0 00015          ADDL2   COUNT, 12(R0)                     ; 2732
                        61      08    AC  C2 0001A          SUBL2   COUNT, (R1)                       ; 2733
                                      50  12 0001E          BNEQ    3$                                ; 2734
                        50      04    BC  0F 00020          REMQUE  @ACB, DUMMY                       ; 2740
                                04    AC  DD 00024          PUSHL   ACB                               ; 2741
                                      10  DD 00027          PUSHL   #16                               ;
                00000000G       00        02  FB 00029          CALLS   #2, FREE_VM                   ;
                                          04  00030          RET                                       ; 2725
            50      0C  A0            61  C1 00031 1$:      ADDL3   (R1), 12(R0), R0                  ; 2744
                        50      08    AC  C2 00036          SUBL2   COUNT, R0                         ;
                    0C  AC              50  D1 0003A          CMPL    R0, LBN                           ;
                                        05  12 0003E          BNEQ    2$                                ;
                        61      08    AC  C2 00040          SUBL2   COUNT, (R1)                       ; 2750
                                          04  00044          RET                                       ; 2744
                                      10  DD 00045 2$:       PUSHL   #16                               ; 2761
                00000000G       00        01  FB 00047          CALLS   #1, GET_VM                    ;
                        04      BC        60  0E 0004E          INSQUE  (NEW), @ACB                   ; 2762
0C  A0              0C    AC    08    AC  C1 00052          ADDL3   COUNT, LBN, 12(NEW)               ; 2763
                        51      04    AC  D0 00059          MOVL    ACB, R1                           ; 2764
            52      0C  A1      08    A1  C1 0005D          ADDL3   8(R1), 12(R1), R2                 ;
08  A0              52  0C      A0    C3 00063          SUBL3   12(NEW), R2, 8(NEW)               ;
08  A1          0C  AC  0C      A1    C3 00069          SUBL3   12(R1), LBN, 8(R1)                ; 2765
                                          04  00070 3$:       RET                                       ; 2767
```

; Routine Size: 113 bytes,   Routine Base: CODE + 049B

```
;  1227          2768  1 %SBTTL 'STA_ALLOC_LBN - allocate specified LBN'
;  1228          2769  1 GLOBAL ROUTINE STA_ALLOC_LBN (COUNT,LBN)=
;  1229          2770  1
;  1230          2771  1 !++
;  1231          2772  1 !
;  1232          2773  1 ! FUNCTIONAL DESCRIPTION:
;  1233          2774  1 !       This routine attempts to allocate an extent at a specific LBN.
;  1234          2775  1 !       No partial allocation is permitted.
;  1235          2776  1 !
;  1236          2777  1 ! INPUT PARAMETERS:
;  1237          2778  1 !       COUNT           - Block count
;  1238          2779  1 !       LBN             - Logical block number
;  1239          2780  1 !
;  1240          2781  1 ! IMPLICIT INPUTS:
;  1241          2782  1 !       CURRENT_VCB
;  1242          2783  1 !
;  1243          2784  1 ! OUTPUT PARAMETERS:
;  1244          2785  1 !       NONE
;  1245          2786  1 !
;  1246          2787  1 ! IMPLICIT OUTPUTS:
;  1247          2788  1 !       NONE
;  1248          2789  1 !
;  1249          2790  1 ! ROUTINE VALUE:
;  1250          2791  1 !       True if the allocation was made, false otherwise.
;  1251          2792  1 !
;  1252          2793  1 ! SIDE EFFECTS:
;  1253          2794  1 !       Allocation list may be modified.
;  1254          2795  1 !
;  1255          2796  1 !--
;  1256          2797  1
;  1257          2798  2 BEGIN
;  1258          2799  2 LOCAL
;  1259          2800  2         ACB:            REF BBLOCK;             ! Pointer to ACB
;  1260          2801  2
;  1261          2802  2
;  1262          2803  2 ACB = .CURRENT_VCB[VCB_ACB_FLINK];
;  1263          2804  2 WHILE .ACB NEQ CURRENT_VCB[VCB_ACB_FLINK] DO
;  1264          2805  3     BEGIN
;  1265          2806  3
;  1266          2807  3     ! If this entry covers the requested extent, allocate it.
;  1267          2808  3     !
;  1268          2809  3     IF
;  1269          2810  3         .LBN GEQU .ACB[ACB_LBN] AND
;  1270          2811  3         .LBN + .COUNT LEQU .ACB[ACB_LBN] + .ACB[ACB_COUNT]
;  1271          2812  3     THEN
;  1272          2813  4         BEGIN
;  1273          2814  4         TAKE_BLOCKS(.ACB, .COUNT, .LBN);
;  1274          2815  4         RETURN TRUE;
;  1275          2816  3         END;
;  1276          2817  3
;  1277          2818  3
;  1278          2819  3     ! If this entry follows the requested extent, exit with failure.
;  1279          2820  3     !
;  1280          2821  3     IF .LBN LSSU .ACB[ACB_LBN]
;  1281          2822  3     THEN
;  1282          2823  3         EXITLOOP;
;  1283          2824  3
```

```
;  1284        2825  3
;  1285        2826  3      ACB = .ACB[ACB_FLINK];
;  1286        2827  2      END;
;  1287        2828  2
;  1288        2829  2
;  1289        2830  2 FALSE
;  1290        2831  1 END;


                                                000C 00000          .ENTRY  STA_ALLOC_LBN, Save R2,R3                       ; 2769
                                    53 00000000'    EF   9E 00002   MOVAB   CURRENT_VCB, R3                                 ;
                                    50              63   D0 00009   MOVL    CURRENT_VCB, R0                                 ; 2803
                                    52        28    A0   D0 0000C   MOVL    40(R0), ACB                                     ;
                          50        63              28   C1 00010 1$:  ADDL3   #40, CURRENT_VCB, R0                           ; 2804
                                    50              52   D1 00014   CMPL    ACB, R0                                         ;
                                                    33   13 00017   BEQL    3$                                              ;
                                0C  A2        08    AC   D1 00019   CMPL    LBN, 12(ACB)                                    ; 2810
                                                    20   1F 0001E   BLSSU   2$                                              ;
                          51    08  AC        04    AC   C1 00020   ADDL3   COUNT, LBN, R1                                   ; 2811
                          50    0C  A2        08    A2   C1 00026   ADDL3   8(ACB), 12(ACB), R0                             ;
                                    50              51   D1 0002C   CMPL    R1, R0                                          ;
                                                    0F   1A 0002F   BGTRU   2$                                              ;
                                    7E        04    AC   7D 00031   MOVQ    COUNT, -(SP)                                    ; 2814
                                                    52   DD 00035   PUSHL   ACB                                             ;
                              FF53  CF              03   FB 00037   CALLS   #3, TAKE_BLOCKS                                 ;
                                    50              01   D0 0003C   MOVL    #1, R0                                          ; 2815
                                                    04   0003F     RET                                                     ;
                                0C  A2        08    AC   D1 00040 2$:  CMPL    LBN, 12(ACB)                                    ; 2821
                                                    05   1F 00045   BLSSU   3$                                              ;
                                    52              62   D0 00047   MOVL    (ACB), ACB                                      ; 2826
                                                    C4   11 0004A   BRB     1$                                              ; 2804
                                                    50   D4 0004C 3$:  CLRL    R0                                              ; 2831
                                                    04   0004E     RET                                                     ;

; Routine Size:  79 bytes,    Routine Base:  CODE + 050C
```

```
;  1292     2832  1 %SBTTL 'STA_ALLOC_BEST - allocate blocks best fit'
;  1293     2833  1 ROUTINE STA_ALLOC_BEST (RCOUNT,ACOUNT,LBN)=
;  1294     2834  1
;  1295     2835  1 !++
;  1296     2836  1 !
;  1297     2837  1 ! FUNCTIONAL DESCRIPTION:
;  1298     2838  1 !       This routine attempts a best fit allocation of an extent.
;  1299     2839  1 !       Partial allocations are allowed.
;  1300     2840  1 !
;  1301     2841  1 ! INPUT PARAMETERS:
;  1302     2842  1 !       RCOUNT          - Requested block count.
;  1303     2843  1 !       ACOUNT          - Pointer to where actual block count is stored.
;  1304     2844  1 !       LBN             - Pointer to where logical block number is stored.
;  1305     2845  1 !
;  1306     2846  1 ! IMPLICIT INPUTS:
;  1307     2847  1 !       CURRENT_VCB
;  1308     2848  1 !
;  1309     2849  1 ! OUTPUT PARAMETERS:
;  1310     2850  1 !       NONE
;  1311     2851  1 !
;  1312     2852  1 ! IMPLICIT OUTPUTS:
;  1313     2853  1 !       NONE
;  1314     2854  1 !
;  1315     2855  1 ! ROUTINE VALUE:
;  1316     2856  1 !       True if the allocation was made, false otherwise.
;  1317     2857  1 !
;  1318     2858  1 ! SIDE EFFECTS:
;  1319     2859  1 !       Allocation list may be modified.
;  1320     2860  1 !
;  1321     2861  1 !--
;  1322     2862  1
;  1323     2863  2 BEGIN
;  1324     2864  2 LOCAL
;  1325     2865  2         RRCOUNT,                                ! Rounded RCOUNT
;  1326     2866  2         MINACB:         REF BBLOCK,             ! Smallest count larger than RCOUNT
;  1327     2867  2         MAXACB:         REF BBLOCK,             ! Largest count
;  1328     2868  2         ACB:            REF BBLOCK;             ! Pointer to ACB
;  1329     2869  2
;  1330     2870  2
;  1331     2871  2 RRCOUNT = (.RCOUNT + .CURRENT_VCB[VCB_CLUSTER] - 1) / .CURRENT_VCB[VCB_CLUSTER] * .CURRENT_VCB[VCB_CLUSTER];
;  1332     2872  2 ACB = .CURRENT_VCB[VCB_ACB_FLINK];
;  1333     2873  2 MINACB = MAXACB = 0;
;  1334     2874  2 WHILE .ACB NEQ CURRENT_VCB[VCB_ACB_FLINK] DO
;  1335     2875  3     BEGIN
;  1336     2876  3
;  1337     2877  3     ! Establish the smallest extent at least as large as the request,
;  1338     2878  3     ! if one exists.
;  1339     2879  3     !
;  1340     2880  3     IF .ACB[ACB_COUNT] GEQU .RRCOUNT
;  1341     2881  3     THEN
;  1342     2882  3         IF .MINACB EQL 0
;  1343     2883  3         THEN
;  1344     2884  3             MINACB = .ACB
;  1345     2885  3         ELSE
;  1346     2886  3             IF .ACB[ACB_COUNT] LSSU .MINACB[ACB_COUNT]
;  1347     2887  3             THEN
;  1348     2888  3                 MINACB = .ACB;
```

```
;  1349          2889  3
;  1350          2890  3
;  1351          2891  3      ! Establish the largest extent.
;  1352          2892  3      !
;  1353          2893  3      IF .MAXACB EQL 0
;  1354          2894  3      THEN
;  1355          2895  3          MAXACB = .ACB
;  1356          2896  3      ELSE
;  1357          2897  3          IF .ACB[ACB_COUNT] GTRU .MAXACB[ACB_COUNT]
;  1358          2898  3          THEN
;  1359          2899  3              MAXACB = .ACB;
;  1360          2900  3
;  1361          2901  3
;  1362          2902  3      ACB = .ACB[ACB_FLINK];
;  1363          2903  2      END;
;  1364          2904  2
;  1365          2905  2
;  1366          2906  2  IF .MINACB NEQ 0
;  1367          2907  2  THEN
;  1368          2908  3      BEGIN
;  1369          2909  3      !
;  1370          2910  3      ! Allocation completely satisfied from the smallest free extent larger
;  1371          2911  3      ! than the request.  Blocks allocated from the beginning of the extent.
;  1372          2912  3      !
;  1373          2913  3      .ACOUNT = .RRCOUNT;
;  1374          2914  3      .LBN = .MINACB[ACB_LBN];
;  1375          2915  3      TAKE_BLOCKS(.MINACB, .RRCOUNT, .MINACB[ACB_LBN]);
;  1376          2916  3      TRUE
;  1377          2917  3      END
;  1378          2918  2  ELSE IF .MAXACB NEQ 0
;  1379          2919  2  THEN
;  1380          2920  3      BEGIN
;  1381          2921  3      !
;  1382          2922  3      ! Allocation partially satisfied from the entire largest free extent.
;  1383          2923  3      !
;  1384          2924  3      .ACOUNT = .MAXACB[ACB_COUNT];
;  1385          2925  3      .LBN = .MAXACB[ACB_LBN];
;  1386          2926  3      TAKE_BLOCKS(.MAXACB, .MAXACB[ACB_COUNT], .MAXACB[ACB_LBN]);
;  1387          2927  3      TRUE
;  1388          2928  3      END
;  1389          2929  2  ELSE
;  1390          2930  3      BEGIN
;  1391          2931  3      .ACOUNT = 0;
;  1392          2932  3      FALSE
;  1393          2933  3      END
;  1394          2934  1  END;


                                003C 00000 STA_ALLOC_BEST:
                                           .WORD   Save R2,R3,R4,R5                        ; 2833
51 00000000'  EF  D0 00002                 MOVL    CURRENT_VCB, R1                         ; 2871
50        04  A1  3C 00009                 MOVZWL  4(R1), R0                               ;
50        04  AC  C0 0000D                 ADDL2   RCOUNT, R0                              ;
              50  D7 00011                 DECL    R0                                      ;
```

```
                                52      04    A1  3C 00013          MOVZWL  4(R1), R2                       ;
                                50            52  C6 00017          DIVL2   R2, R0                          ;
                                55      04    A1  3C 0001A          MOVZWL  4(R1), RRCOUNT                  ;
                                55            50  C4 0001E          MULL2   R0, RRCOUNT                     ;
                                50      28    A1  D0 00021          MOVL    40(R1), ACB                     ; 2872
                                              52  7C 00025          CLRQ    MAXACB                          ; 2873
                                54      28    A1  9E 00027 1$:      MOVAB   40(R1), R4                      ; 2874
                                54            50  D1 0002B          CMPL    ACB, R4                         ;
                                              27  13 0002E          BEQL    6$                              ;
                                55      08    A0  D1 00030          CMPL    8(ACB), RRCOUNT                 ; 2880
                                              0E  1F 00034          BLSSU   3$                              ;
                                              53  D5 00036          TSTL    MINACB                          ; 2882
                                              07  13 00038          BEQL    2$                              ;
                          08    A3      08    A0  D1 0003A          CMPL    8(ACB), 8(MINACB)               ; 2886
                                              03  1E 0003F          BGEQU   3$                              ;
                                53            50  D0 00041 2$:      MOVL    ACB, MINACB                     ; 2888
                                              52  D5 00044 3$:      TSTL    MAXACB                          ; 2893
                                              07  13 00046          BEQL    4$                              ;
                          08    A2      08    A0  D1 00048          CMPL    8(ACB), 8(MAXACB)               ; 2897
                                              03  1B 0004D          BLEQU   5$                              ;
                                52            50  D0 0004F 4$:      MOVL    ACB, MAXACB                     ; 2899
                                50            60  D0 00052 5$:      MOVL    (ACB), ACB                      ; 2902
                                              D0  11 00055          BRB     1$                              ; 2874
                                              52  D5 00057 6$:      TSTL    MINACB                          ; 2906
                                              10  13 00059          BEQL    7$                              ;
                          08    BC            55  D0 0005B          MOVL    RRCOUNT, @ACOUNT                ; 2913
                          0C    BC      0C    A3  D0 0005F          MOVL    12(MINACB), @LBN                ; 2914
                                0C      0C    A3  DD 00064          PUSHL   12(MINACB)                      ; 2915
                                              28  BB 00067          PUSHR   #^M<R3,R5>                      ;
                                              14  11 00069          BRB     8$                              ;
                                              52  D5 0006B 7$:      TSTL    MAXACB                          ; 2918
                                              19  13 0006D          BEQL    9$                              ;
                          08    BC      08    A2  D0 0006F          MOVL    8(MAXACB), @ACOUNT              ; 2924
                          0C    BC      0C    A2  D0 00074          MOVL    12(MAXACB), @LBN                ; 2925
                                7E      08    A2  7D 00079          MOVQ    8(MAXACB), -(SP)                ; 2926
                                              52  DD 0007D          PUSHL   MAXACB                          ;
                        FEBC    CF            03  FB 0007F 8$:      CALLS   #3, TAKE_BLOCKS                 ;
                                50            01  D0 00084          MOVL    #1, R0                          ; 2920
                                              04  00087          RET                                     ;
                                          08  BC  D4 00088 9$:      CLRL    @ACOUNT                         ; 2931
                                              50  D4 0008B          CLRL    R0                              ; 2930
                                                  04 0008D          RET                                     ; 2934

; Routine Size:  142 bytes,    Routine Base:  CODE + 055B
```

```
;  1396     2935  1 %SBTTL 'FREE_BLOCKS - return blocks to free list'
;  1397     2936  1 GLOBAL ROUTINE FREE_BLOCKS (COUNT,LBN): NOVALUE=
;  1398     2937  1
;  1399     2938  1 !++
;  1400     2939  1 !
;  1401     2940  1 ! FUNCTIONAL DESCRIPTION:
;  1402     2941  1 !       This routine puts the specified blocks back onto the free list
;  1403     2942  1 !       for the currently active volume.
;  1404     2943  1 !
;  1405     2944  1 ! INPUT PARAMETERS:
;  1406     2945  1 !       COUNT           - Count of blocks to be freed.
;  1407     2946  1 !       LBN             - Starting LBN.
;  1408     2947  1 !
;  1409     2948  1 ! IMPLICIT INPUTS:
;  1410     2949  1 !       CURRENT_VCB     - Pointer to VCB for volume.
;  1411     2950  1 !
;  1412     2951  1 ! OUTPUT PARAMETERS:
;  1413     2952  1 !       NONE
;  1414     2953  1 !
;  1415     2954  1 ! IMPLICIT OUTPUTS:
;  1416     2955  1 !       NONE
;  1417     2956  1 !
;  1418     2957  1 ! ROUTINE VALUE:
;  1419     2958  1 !       NONE
;  1420     2959  1 !
;  1421     2960  1 ! SIDE EFFECTS:
;  1422     2961  1 !       ACB list altered.
;  1423     2962  1 !
;  1424     2963  1 !--
;  1425     2964  1
;  1426     2965  2 BEGIN
;  1427     2966  2 LOCAL
;  1428     2967  2         ACB:            REF BBLOCK,     ! ACB following released extent
;  1429     2968  2         PREV_ACB:       REF BBLOCK,     ! ACB preceding released extent
;  1430     2969  2         ALLOC_ACB:      REF BBLOCK;     ! ACB allocated or deallocated
;  1431     2970  2
;  1432     2971  2
;  1433     2972  2 ! Protect against bogus parameters.
;  1434     2973  2 !
;  1435     2974  2 IF .COUNT EQL 0 THEN RETURN;
;  1436     2975  2
;  1437     2976  2
;  1438     2977  2 ! Search the ACB list on the appropriate VCB for the ACB describing an extent
;  1439     2978  2 ! following the extent to be released.  The ACB list is in LBN order.
;  1440     2979  2 !
;  1441     2980  2 ACB = CURRENT_VCB[VCB_ACB_FLINK];
;  1442     2981  2 DO ACB = .ACB[ACB_FLINK]
;  1443     2982  2 UNTIL .ACB EQL CURRENT_VCB[VCB_ACB_FLINK] OR .ACB[ACB_LBN] GTRU .LBN;
;  1444     2983  2 PREV_ACB = .ACB[ACB_BLINK];
;  1445     2984  2
;  1446     2985  2
;  1447     2986  2 ! Check for contiguity with ACB or PREV_ACB or both and proceed accordingly.
;  1448     2987  2 !
;  1449     2988  2 IF
;  1450     2989  2     .PREV_ACB NEQ CURRENT_VCB[VCB_ACB_FLINK] AND
;  1451     2990  2     .PREV_ACB[ACB_LBN] + .PREV_ACB[ACB_COUNT] EQL .LBN
;  1452     2991  2 THEN
```

```
;  1453          2992  2     IF
;  1454          2993  2         .ACB NEQ CURRENT_VCB[VCB_ACB_FLINK] AND
;  1455          2994  2         .LBN + .COUNT EQL .ACB[ACB_LBN]
;  1456          2995  2     THEN
;  1457          2996  3         BEGIN
;  1458          2997  3         !
;  1459          2998  3         ! Contiguous with both.  Modify PREV_ACB to describe all 3 extents and
;  1460          2999  3         ! release ACB.
;  1461          3000  3         !
;  1462          3001  3         PREV_ACB[ACB_COUNT] = .PREV_ACB[ACB_COUNT] + .COUNT + .ACB[ACB_COUNT];
;  1463          3002  3         REMQUE(.PREV_ACB, ALLOC_ACB);
;  1464          3003  3         FREE_VM(ACB_S_ENTRY, .ALLOC_ACB);
;  1465          3004  3         END
;  1466          3005  2     ELSE
;  1467          3006  3         BEGIN
;  1468          3007  3         !
;  1469          3008  3         ! Contiguous with PREV_ACB but not ACB.  Modify PREV_ACB to describe
;  1470          3009  3         ! both extents.
;  1471          3010  3         !
;  1472          3011  3         PREV_ACB[ACB_COUNT] = .PREV_ACB[ACB_COUNT] + .COUNT;
;  1473          3012  3         END
;  1474          3013  2 ELSE
;  1475          3014  2     IF
;  1476          3015  2         .ACB NEQ CURRENT_VCB[VCB_ACB_FLINK] AND
;  1477          3016  2         .LBN + .COUNT EQL .ACB[ACB_LBN]
;  1478          3017  2     THEN
;  1479          3018  3         BEGIN
;  1480          3019  3         !
;  1481          3020  3         ! Contiguous with ACB but not PREV_ACB.  Modify ACB to describe
;  1482          3021  3         ! both extents.
;  1483          3022  3         !
;  1484          3023  3         ACB[ACB_LBN] = .LBN;
;  1485          3024  3         ACB[ACB_COUNT] = .ACB[ACB_COUNT] + .COUNT;
;  1486          3025  3         END
;  1487          3026  2     ELSE
;  1488          3027  3         BEGIN
;  1489          3028  3         !
;  1490          3029  3         ! Contiguous with neither.  Generate a new ACB.
;  1491          3030  3         !
;  1492          3031  3         ALLOC_ACB = GET_VM(ACB_S_ENTRY);
;  1493          3032  3         INSQUE(.ALLOC_ACB, .PREV_ACB);
;  1494          3033  3         ALLOC_ACB[ACB_COUNT] = .COUNT;
;  1495          3034  3         ALLOC_ACB[ACB_LBN] = .LBN;
;  1496          3035  2         END;
;  1497          3036  1 END;


                                    001C 00000         .ENTRY  FREE_BLOCKS, Save R2,R3,R4                  ; 2936
                        54      04  AC  D0 00002        MOVL    COUNT, R4                                   ; 2974
                                    01  12 00006        BNEQ    1$                                          ;
                                        04 00008        RET                                                 ;
             53 00000000'       EF      28  C1 00009 1$:   ADDL3   #40, CURRENT_VCB, R3                    ; 2980
                        50          53  D0 00011        MOVL    R3, ACB                                     ;
                        50          60  D0 00014 2$:    MOVL    (ACB), ACB                                  ; 2981
```

```
                                53              50  D1 00017          CMPL    ACB, R3                                 ; 2982
                                                07  13 0001A          BEQL    3$                                      ;
                            08  AC          0C  A0  D1 0001C          CMPL    12(ACB), LBN                            ;
                                                F1  1B 00021          BLEQU   2$                                      ;
                                52          04  A0  D0 00023 3$:      MOVL    4(ACB), PREV_ACB                        ; 2983
                                53              52  D1 00027          CMPL    PREV_ACB, R3                            ; 2989
                                                3B  13 0002A          BEQL    5$                                      ;
                        51      0C          A2  08  A2  C1 0002C      ADDL3   8(PREV_ACB), 12(PREV_ACB), R1           ; 2990
                            08  AC              51  D1 00032          CMPL    R1, LBN                                 ;
                                                2F  12 00036          BNEQ    5$                                      ;
                                53              50  D1 00038          CMPL    ACB, R3                                 ; 2993
                                                25  13 0003B          BEQL    4$                                      ;
                        51                  54  08  AC  C1 0003D      ADDL3   LBN, R4, R1                             ; 2994
                            0C  A0              51  D1 00042          CMPL    R1, 12(ACB)                             ;
                                                1A  12 00046          BNEQ    4$                                      ;
                        51                  54  08  A2  C1 00048      ADDL3   8(PREV_ACB), R4, R1                     ; 3001
                            08  A2          08  B041  9E 0004D        MOVAB   @8(ACB)[R1], 8(PREV_ACB)                ;
                                53              62  0F 00053          REMQUE  (PREV_ACB), ALLOC_ACB                   ; 3002
                                                53  DD 00056          PUSHL   ALLOC_ACB                               ; 3003
                                                10  DD 00058          PUSHL   #16                                     ;
                    00000000G   00              02  FB 0005A          CALLS   #2, FREE_VM                             ;
                                                04  00061          RET                                             ; 2992
                            08  A2              54  C0 00062 4$:      ADDL2   R4, 8(PREV_ACB)                         ; 3011
                                                04  00066          RET                                             ; 2992
                                53              50  D1 00067 5$:      CMPL    ACB, R3                                 ; 3015
                                                15  13 0006A          BEQL    6$                                      ;
                        51                  54  08  AC  C1 0006C      ADDL3   LBN, R4, R1                             ; 3016
                            0C  A0              51  D1 00071          CMPL    R1, 12(ACB)                             ;
                                                0A  12 00075          BNEQ    6$                                      ;
                            0C  A0          08  AC  D0 00077          MOVL    LBN, 12(ACB)                            ; 3023
                            08  A0              54  C0 0007C          ADDL2   R4, 8(ACB)                              ; 3024
                                                04  00080          RET                                             ; 3014
                                                10  DD 00081 6$:      PUSHL   #16                                     ; 3031
                    00000000G   00              01  FB 00083          CALLS   #1, GET_VM                              ;
                                53              50  D0 0008A          MOVL    R0, ALLOC_ACB                           ;
                                62              63  0E 0008D          INSQUE  (ALLOC_ACB), (PREV_ACB)                 ; 3032
                            08  A3          04  AC  7D 00090          MOVQ    COUNT, 8(ALLOC_ACB)                     ; 3033
                                                04  00095          RET                                             ; 3036

; Routine Size:  150 bytes,    Routine Base:  CODE + 05E9
```

```
; 1499          3037  1 %SBTTL 'MAKE_POINTER1 - make ODS-1 map pointer'
; 1500          3038  1 GLOBAL ROUTINE MAKE_POINTER1 (BUFFER,COUNT,LBN,UNMAPPED)=
; 1501          3039  1
; 1502          3040  1 !++
; 1503          3041  1 !
; 1504          3042  1 ! FUNCTIONAL DESCRIPTION:
; 1505          3043  1 !       This routine appends retrieval pointers to the map area of an
; 1506          3044  1 !       ODS-1 file header describing the given count and LBN.
; 1507          3045  1 !
; 1508          3046  1 ! INPUT PARAMETERS:
; 1509          3047  1 !       BUFFER          - Pointer to file header buffer
; 1510          3048  1 !       COUNT           - Block count
; 1511          3049  1 !       LBN             - Starting logical block number
; 1512          3050  1 !       UNMAPPED        - (optional) Number of blocks not mapped
; 1513          3051  1 !
; 1514          3052  1 ! IMPLICIT INPUTS:
; 1515          3053  1 !       NONE
; 1516          3054  1 !
; 1517          3055  1 ! OUTPUT PARAMETERS:
; 1518          3056  1 !       NONE
; 1519          3057  1 !
; 1520          3058  1 ! IMPLICIT OUTPUTS:
; 1521          3059  1 !       NONE
; 1522          3060  1 !
; 1523          3061  1 ! ROUTINE VALUE:
; 1524          3062  1 !       SS$_NORMAL or SS$_HEADERFULL.
; 1525          3063  1 !
; 1526          3064  1 ! SIDE EFFECTS:
; 1527          3065  1 !       NONE
; 1528          3066  1 !
; 1529          3067  1 !--
; 1530          3068  1
; 1531          3069  2 BEGIN
; 1532          3070  2
; 1533          3071  2 BUILTIN
; 1534          3072  2         ACTUALCOUNT;
; 1535          3073  2
; 1536          3074  2 MAP
; 1537          3075  2         BUFFER:         REF BBLOCK;     ! Pointer to file header buffer
; 1538          3076  2 LOCAL
; 1539          3077  2         CURRENT_COUNT,                  ! running block count
; 1540          3078  2         CURRENT_LBN,                    ! running LBN
; 1541          3079  2         MAP_AREA:       REF BBLOCK,     ! pointer to map area
; 1542          3080  2         MAP_POINTER:    REF BBLOCK;     ! pointer to map area
; 1543          3081  2
; 1544          3082  2
; 1545          3083  2 ! Compute the address in the file header where the pointer should go.
; 1546          3084  2 ! Then determine the format of the pointer and build it.
; 1547          3085  2 !
; 1548          3086  2 MAP_AREA = .BUFFER + 2 * .BUFFER[FH2$B_MPOFFSET];
; 1549          3087  2 MAP_POINTER = .MAP_AREA + FM1$C_POINTERS + .MAP_AREA[FM1$B_INUSE]*2;
; 1550          3088  2 CURRENT_COUNT = .COUNT;
; 1551          3089  2 CURRENT_LBN = .LBN;
; 1552          3090  2
; 1553          3091  2
; 1554          3092  2 DO
; 1555          3093  3     BEGIN
```

```
;  1556          3094  3
;  1557          3095  3      ! Check for map area overflow.
;  1558          3096  3      !
;  1559          3097  3      IF .MAP_POINTER + 4 GTRA .BUFFER + $BYTEOFFSET(FH1$W_CHECKSUM)
;  1560          3098  3      THEN
;  1561          3099  4          BEGIN
;  1562          3100  4          IF ACTUALCOUNT () GEQU 4
;  1563          3101  4          THEN .UNMAPPED = .CURRENT_COUNT;
;  1564          3102  4          RETURN SS$_HEADERFULL;
;  1565          3103  3          END;
;  1566          3104  3
;  1567          3105  3
;  1568          3106  3      ! Build the map pointer.
;  1569          3107  3      !
;  1570          3108  3      MAP_AREA[FM1$B_INUSE] = .MAP_AREA[FM1$B_INUSE] + 2;
;  1571          3109  3      MAP_POINTER[FM1$B_HIGHLBN] = .CURRENT_LBN<16,8>;
;  1572          3110  3      MAP_POINTER[FM1$B_COUNT] = MIN(.CURRENT_COUNT, 256) - 1;
;  1573          3111  3      MAP_POINTER[FM1$W_LOWLBN] = .CURRENT_LBN<0,16>;
;  1574          3112  3      MAP_POINTER = .MAP_POINTER + 4;
;  1575          3113  3
;  1576          3114  3
;  1577          3115  3      ! Decrease residual count.
;  1578          3116  3      !
;  1579          3117  3      CURRENT_LBN = .CURRENT_LBN + MIN(.CURRENT_COUNT, 256);
;  1580          3118  3      CURRENT_COUNT = .CURRENT_COUNT - MIN(.CURRENT_COUNT, 256);
;  1581          3119  3      END
;  1582          3120  2 UNTIL .CURRENT_COUNT EQL 0;
;  1583          3121  2
;  1584          3122  2
;  1585          3123  2 SS$_NORMAL
;  1586          3124  1 END;


                                        003C 00000          .ENTRY  MAKE_POINTER1, Save R2,R3,R4,R5          ; 3038
                                  53    04   AC  D0 00002          MOVL    BUFFER, R3                       ; 3086
                                  50    01   A3  9A 00006          MOVZBL  1(R3), R0                        ;
                                  52       8340  3E 0000A          MOVAW   (R3)+[R0], MAP_AREA              ;
                                  50    08   A2  9A 0000E          MOVZBL  8(MAP_AREA), R0                  ; 3087
                                  50    0A A240  3E 00012          MOVAW   10(MAP_AREA)[R0], MAP_POINTER    ;
                                  51    08   AC  D0 00017          MOVL    COUNT, CURRENT_COUNT             ; 3088
                                  54    0C   AC  D0 0001B          MOVL    LBN, CURRENT_LBN                 ; 3089
                                  53  01FC   C3  9E 0001F          MOVAB   508(R3), R3                      ; 3097
                                  55    04   A0  9E 00024 1$:      MOVAB   4(R0), R5                        ;
                                  53         55  D1 00028          CMPL    R5, R3                           ;
                                             0F  1B 0002B          BLEQU   3$                               ;
                                  04         6C  91 0002D          CMPB    (AP), #4                         ; 3100
                                             04  1F 00030          BLSSU   2$                               ;
                                10   BC       51  D0 00032          MOVL    CURRENT_COUNT, @UNMAPPED         ; 3101
                                  50  08C8    8F  3C 00036 2$:      MOVZWL  #2248, R0                        ; 3102
                                              04  0003B          RET                                      ;
                                10   A2       02  80 0003C 3$:      ADDB2   #2, 8(MAP_AREA)                  ; 3108
                55                54    08   10  EF 00040          EXTZV   #16, #8, CURRENT_LBN, R5         ; 3109
                                  60         55  90 00045          MOVB    R5, (MAP_POINTER)                ;
                                  55         51  D0 00048          MOVL    CURRENT_COUNT, R5                ; 3110
```

```
                00000100   8F                55  D1 00048          CMPL    R5, #256                                ;
                                             05  15 00052          BLEQ    4$                                      ;
                           55       0100     8F  3C 00054          MOVZWL  #256, R5                                ;
          01    A0                 55         01  83 00059 4$:     SUBB3   #1, R5, 1(MAP_POINTER)                  ;
                      02   A0                54  B0 0005E          MOVW    CURRENT_LBN, 2(MAP_POINTER)             ; 3111
                           50                04  C0 00062          ADDL2   #4, MAP_POINTER                         ; 3112
                           54                55  C0 00065          ADDL2   R5, CURRENT_LBN                         ; 3117
                           51                55  C2 00068          SUBL2   R5, CURRENT_COUNT                       ; 3118
                                             B7  12 0006B          BNEQ    1$                                      ; 3120
                           50                01  D0 0006D          MOVL    #1, R0                                  ; 3124
                                                 04 00070          RET                                             ;

; Routine Size:  113 bytes,    Routine Base:  CODE + 067F
```

```
;1588        3125  1 %SBTTL 'MAKE_POINTER - make ODS-2 map pointer'
;1589        3126  1 GLOBAL ROUTINE MAKE_POINTER (BUFFER,COUNT,LBN)=
;1590        3127  1
;1591        3128  1 !++
;1592        3129  1 !
;1593        3130  1 ! FUNCTIONAL DESCRIPTION:
;1594        3131  1 !       This routine appends a retrieval pointer to the map area of an
;1595        3132  1 !       ODS-2 file header describing the given count and LBN.
;1596        3133  1 !
;1597        3134  1 ! INPUT PARAMETERS:
;1598        3135  1 !       BUFFER          - Pointer to file header buffer
;1599        3136  1 !       COUNT           - Block count
;1600        3137  1 !       LBN             - Starting logical block number
;1601        3138  1 !
;1602        3139  1 ! IMPLICIT INPUTS:
;1603        3140  1 !       NONE
;1604        3141  1 !
;1605        3142  1 ! OUTPUT PARAMETERS:
;1606        3143  1 !       CURRENT_VCB     - Pointer to VCB for selected volume.
;1607        3144  1 !
;1608        3145  1 ! IMPLICIT OUTPUTS:
;1609        3146  1 !       NONE
;1610        3147  1 !
;1611        3148  1 ! ROUTINE VALUE:
;1612        3149  1 !       SS$_NORMAL or SS$_HEADERFULL.
;1613        3150  1 !
;1614        3151  1 ! SIDE EFFECTS:
;1615        3152  1 !       NONE
;1616        3153  1 !
;1617        3154  1 !--
;1618        3155  1
;1619        3156  2 BEGIN
;1620        3157  2 MAP
;1621        3158  2         BUFFER:         REF BBLOCK;     ! Pointer to file header buffer
;1622        3159  2 LOCAL
;1623        3160  2         MAP_POINTER:    REF BBLOCK;     ! pointer to map area
;1624        3161  2
;1625        3162  2
;1626        3163  2 ! Compute the address in the file header where the pointer should go.
;1627        3164  2 ! Then determine the format of the pointer and build it.
;1628        3165  2 !
;1629        3166  2 MAP_POINTER = .BUFFER + 2 * (.BUFFER[FH2$B_MPOFFSET] + .BUFFER[FH2$B_MAP_INUSE]);
;1630        3167  2
;1631        3168  2
;1632        3169  2 IF .COUNT LEQU 256 AND .LBN LSSU 1^22
;1633        3170  2 THEN
;1634        3171  3     BEGIN
;1635        3172  3
;1636        3173  3     ! Check for map area overflow.
;1637        3174  3     !
;1638        3175  3     IF .MAP_POINTER + 4 GTRA .BUFFER + 2 * .BUFFER[FH2$B_ACOFFSET]
;1639        3176  3     THEN
;1640        3177  3         RETURN SS$_HEADERFULL;
;1641        3178  3
;1642        3179  3
;1643        3180  3     ! Build the map pointer.
;1644        3181  3     !
```

```
; 1645          3182  3      MAP_POINTER[FM2$V_FORMAT] = FM2$C_FORMAT1;
; 1646          3183  3      MAP_POINTER[FM2$B_COUNT1] = .COUNT - 1;
; 1647          3184  3      MAP_POINTER[FM2$V_HIGHLBN] = .LBN<16,6>;
; 1648          3185  3      MAP_POINTER[FM2$W_LOWLBN] = .LBN<0,16>;
; 1649          3186  3      BUFFER[FH2$B_MAP_INUSE] = .BUFFER[FH2$B_MAP_INUSE] + 2;
; 1650          3187  3      END
; 1651          3188  2 ELSE IF .COUNT LEQU 16384
; 1652          3189  2 THEN
; 1653          3190  3      BEGIN
; 1654          3191  3
; 1655          3192  3      ! Check for map area overflow.
; 1656          3193  3      !
; 1657          3194  3      IF .MAP_POINTER + 6 GTRA .BUFFER + 2 * .BUFFER[FH2$B_ACOFFSET]
; 1658          3195  3      THEN
; 1659          3196  3          RETURN SS$_HEADERFULL;
; 1660          3197  3
; 1661          3198  3
; 1662          3199  3      ! Build the map pointer.
; 1663          3200  3      !
; 1664          3201  3      MAP_POINTER[FM2$V_FORMAT] = FM2$C_FORMAT2;
; 1665          3202  3      MAP_POINTER[FM2$V_COUNT2] = .COUNT - 1;
; 1666          3203  3      MAP_POINTER[FM2$L_LBN2] = .LBN;
; 1667          3204  3      BUFFER[FH2$B_MAP_INUSE] = .BUFFER[FH2$B_MAP_INUSE] + 3;
; 1668          3205  3      END
; 1669          3206  2 ELSE IF .COUNT LEQU 1^30
; 1670          3207  2 THEN
; 1671          3208  3      BEGIN
; 1672          3209  3
; 1673          3210  3      ! Check for map area overflow.
; 1674          3211  3      !
; 1675          3212  3      IF .MAP_POINTER + 8 GTRA .BUFFER + 2 * .BUFFER[FH2$B_ACOFFSET]
; 1676          3213  3      THEN
; 1677          3214  3          RETURN SS$_HEADERFULL;
; 1678          3215  3
; 1679          3216  3
; 1680          3217  3      ! Build the map pointer.
; 1681          3218  3      !
; 1682          3219  3      .MAP_POINTER = ROT(.COUNT-1, 16);
; 1683          3220  3      MAP_POINTER[FM2$V_FORMAT] = FM2$C_FORMAT3;
; 1684          3221  3      MAP_POINTER[FM2$L_LBN3] = .LBN;
; 1685          3222  3      BUFFER[FH2$B_MAP_INUSE] = .BUFFER[FH2$B_MAP_INUSE] + 4;
; 1686          3223  3      END
; 1687          3224  2 ELSE
; 1688          3225  2      SIGNAL(BACKUP$_LARGECNT, 1, CURRENT_VCB[VCB_DEVICE]);
; 1689          3226  2
; 1690          3227  2
; 1691          3228  2 SS$_NORMAL
; 1692          3229  1 END;


                                    001C 00000          .ENTRY  MAKE_POINTER, Save R2,R3,R4                     ; 3126
                51      04    AC   D0 00002          MOVL    BUFFER, R1                                      ; 3166
                50      01    A1   9A 00006          MOVZBL  1(R1), R0                                       ;
                52      3A    A1   9A 0000A          MOVZBL  58(R1), R2                                      ;
```

```
                                50              52  C0 0000E            ADDL2   R2, R0                                  ;
                                50          6140  3E 00011            MOVAW   (R1)[R0], MAP_POINTER                    ;
                                53       08   AC  D0 00015            MOVL    COUNT, R3                                ; 3169
                    00000100    8F            53  D1 00019            CMPL    R3, #256                                 ;
                                              36  1A 00020            BGTRU   1$                                       ;
                    00400000    8F       0C   AC  D1 00022            CMPL    LBN, #4194304                            ;
                                              2C  1E 0002A            BGEQU   1$                                       ;
                                54       04   A0  9E 0002C            MOVAB   4(R0), R4                                ; 3175
                                52       02   A1  9A 00030            MOVZBL  2(R1), R2                                ;
                                52          6142  3E 00034            MOVAW   (R1)[R2], R2                             ;
                                52            54  D1 00038            CMPL    R4, R2                                   ;
                                              68  1A 0003B            BGTRU   3$                                       ;
          60        02                    0E      01  F0 0003D            INSV    #1, #14, #2, (MAP_POINTER)               ; 3182
          60                              53      01  83 00042            SUBB3   #1, R3, (MAP_POINTER)                    ; 3183
    01    A0        06                    00   0E   AC  F0 00046            INSV    LBN+2, #0, #8, 1(MAP_POINTER)            ; 3184
                          02            A0   0C   AC  B0 0004D            MOVW    LBN, 2(MAP_POINTER)                      ; 3185
                          3A            A1        02  80 00052            ADDB2   #2, 58(R1)                               ; 3186
                                              69  11 00056            BRB     5$                                       ; 3169
                    00004000    8F            53  D1 00058 1$:        CMPL    R3, #16384                               ; 3188
                                              2A  1A 0005F            BGTRU   2$                                       ;
                                54       06   A0  9E 00061            MOVAB   6(R0), R4                                ; 3194
                                52       02   A1  9A 00065            MOVZBL  2(R1), R2                                ;
                                52          6142  3E 00069            MOVAW   (R1)[R2], R2                             ;
                                52            54  D1 0006D            CMPL    R4, R2                                   ;
                                              33  1A 00070            BGTRU   3$                                       ;
          60        02                    0E      02  F0 00072            INSV    #2, #14, #2, (MAP_POINTER)               ; 3201
                                52       FF   A3  9E 00077            MOVAB   -1(R3), R2                               ; 3202
          60        0E                    00      52  F0 0007B            INSV    R2, #0, #14, (MAP_POINTER)               ;
                          02            A0   0C   AC  D0 00080            MOVL    LBN, 2(MAP_POINTER)                      ; 3203
                          3A            A1        03  80 00085            ADDB2   #3, 58(R1)                               ; 3204
                                              4F  11 00089            BRB     7$                                       ; 3188
                    40000000    8F            53  D1 0008B 2$:        CMPL    R3, #1073741824                          ; 3206
                                              2F  1A 00092            BGTRU   6$                                       ;
                                54       08   A0  9E 00094            MOVAB   8(R0), R4                                ; 3212
                                52       02   A1  9A 00098            MOVZBL  2(R1), R2                                ;
                                52          6142  3E 0009C            MOVAW   (R1)[R2], R2                             ;
                                52            54  D1 000A0            CMPL    R4, R2                                   ;
                                              06  1B 000A3            BLEQU   4$                                       ;
                                50     08C8   8F  3C 000A5 3$:        MOVZWL  #2248, R0                                ; 3214
                                                  04 000AA            RET                                              ;
                                52       FF   A3  9E 000AB 4$:        MOVAB   -1(R3), R2                               ; 3219
          60                    52            10  9C 000AF            ROTL    #16, R2, (MAP_POINTER)                   ;
                          01    A0       C0   8F  88 000B3            BISB2   #192, 1(MAP_POINTER)                     ; 3220
                          04    A0       0C   AC  D0 000B8            MOVL    LBN, 4(MAP_POINTER)                      ; 3221
                          3A    A1            04  80 000BD            ADDB2   #4, 58(R1)                               ; 3222
                                              17  11 000C1 5$:        BRB     7$                                       ; 3206
          7E 00000000'  EF                    20  C1 000C3 6$:        ADDL3   #32, CURRENT_VCB, -(SP)                  ; 3225
                                              01  DD 000CB            PUSHL   #1                                       ;
                                00000000G  8F   DD 000CD            PUSHL   #BACKUP$_LARGECNT                        ;
                    00000000G   00              03  FB 000D3            CALLS   #3, LIB$SIGNAL                           ;
                                50              01  D0 000DA 7$:        MOVL    #1, R0                                   ; 3229
                                                    04 000DD            RET                                              ;

; Routine Size:  222 bytes,    Routine Base:  CODE + 06F0
```

```
;1694        3230  1 %SBTTL 'CREATE_WINDOW - create a window block'
;1695        3231  1 GLOBAL ROUTINE CREATE_WINDOW (P_HEADER,P_RVN,P_WINDOW,START_VBN,WINDOW_SIZE)=
;1696        3232  1
;1697        3233  1 !++
;1698        3234  1 !
;1699        3235  1 ! FUNCTIONAL DESCRIPTION:
;1700        3236  1 !       This routine generates a window block (or blocks) from a file
;1701        3237  1 !       header, reading the extension headers as necessary.
;1702        3238  1 !
;1703        3239  1 ! INPUT PARAMETERS:
;1704        3240  1 !       P_HEADER        - Pointer to file header to be processed.
;1705        3241  1 !       P_RVN           - Relative volume number of file header.
;1706        3242  1 !       P_WINDOW        - Pointer to where window is returned.
;1707        3243  1 !       START_VBN       - Starting VBN of file header.
;1708        3244  1 !       WINDOW_SIZE     - Minimum size to allocate for window in pointers.
;1709        3245  1 !
;1710        3246  1 ! IMPLICIT INPUTS:
;1711        3247  1 !       CURRENT_MTL     - Pointer to MTL for selected volume set.
;1712        3248  1 !
;1713        3249  1 ! OUTPUT PARAMETERS:
;1714        3250  1 !       NONE
;1715        3251  1 !
;1716        3252  1 ! IMPLICIT OUTPUTS:
;1717        3253  1 !       CURRENT_MTL[MTL_FILESIZE] - Contains total space allocated to file.
;1718        3254  1 !
;1719        3255  1 ! ROUTINE VALUE:
;1720        3256  1 !       Completion status.
;1721        3257  1 !
;1722        3258  1 ! SIDE EFFECTS:
;1723        3259  1 !       NONE
;1724        3260  1 !
;1725        3261  1 !--
;1726        3262  1
;1727        3263  2 BEGIN
;1728        3264  2 LINKAGE
;1729        3265  2         L_MAP_POINTER=  JSB:
;1730        3266  2                         GLOBAL(COUNT=6, LBN=7, MAP_POINTER=8);
;1731        3267  2 EXTERNAL ROUTINE
;1732        3268  2         GET_MAP_POINTER: L_MAP_POINTER; ! Get value of ODS-2 file map pointer
;1733        3269  2 LOCAL
;1734        3270  2         STATUS,                         ! Status return
;1735        3271  2         W_SIZE,                         ! size to allocate window
;1736        3272  2         DYNWCB:         REF BBLOCK,     ! Pointer to dynamic WCB
;1737        3273  2         HEADER:         REF BBLOCK,     ! Pointer to current file header
;1738        3274  2         RVN,                            ! RVN of current file header
;1739        3275  2         EXT_FILE_ID:    BBLOCK[FID$C_LENGTH],   ! Extension file ID
;1740        3276  2         LOCAL_HEADER:   BBLOCK[512],    ! Local area for file header
;1741        3277  2         LAST_WINDOW:    REF BBLOCK,     ! Last block in window list
;1742        3278  2         WINDOW:         BBLOCK[WCB_S_HEADER + 255 * WCB_S_ENTRY],
;1743        3279  2         P:              REF BBLOCK;     ! Pointer to current window entry
;1744        3280  2
;1745        3281  2
;1746        3282  2 ! Initialize.
;1747        3283  2 !
;1748        3284  2 .P_WINDOW = 0;
;1749        3285  2 CURRENT_MTL[MTL_FILESIZE] = 0;
;1750        3286  2 HEADER = .P_HEADER;
```

```
;  1751          3287  2 RVN = .P_RVN;
;  1752          3288  2 LAST_WINDOW = 0;
;  1753          3289  2 WINDOW[WCB_LINK] = 0;
;  1754          3290  2 WINDOW[WCB_VBN] = .START_VBN;
;  1755          3291  2 WINDOW[WCB_SIZE] = 0;
;  1756          3292  2 WINDOW[WCB_RVN] = .RVN;
;  1757          3293  2 WINDOW[WCB_FLAGS] = 0;
;  1758          3294  2 P = WINDOW + WCB_S_HEADER - WCB_S_ENTRY;
;  1759          3295  2 IF .HEADER[FH2$B_STRUCLEV] EQL 2
;  1760          3296  2 AND .HEADER[FH2$W_SEG_NUM] EQL 0
;  1761          3297  2 AND .HEADER[FH2$B_IDOFFSET] GEQU ($BYTEOFFSET (FH2$L_HIGHWATER) + 4) / 2
;  1762          3298  2 AND .HEADER[FH2$L_HIGHWATER] NEQ 0
;  1763          3299  2 THEN
;  1764          3300  3     BEGIN
;  1765          3301  3     WINDOW[WCB_CUR_HWM] = .HEADER[FH2$L_HIGHWATER];
;  1766          3302  3     WINDOW[WCB_SET_HWM] = .HEADER[FH2$L_HIGHWATER];
;  1767          3303  3     END
;  1768          3304  2 ELSE
;  1769          3305  3     BEGIN
;  1770          3306  3     WINDOW[WCB_CUR_HWM] = -1;
;  1771          3307  3     WINDOW[WCB_SET_HWM] = -1;
;  1772          3308  2     END;
;  1773          3309  2
;  1774          3310  2
;  1775          3311  2 ! Loop over this header and all of its extension headers.
;  1776          3312  2 !
;  1777          3313  2 WHILE TRUE DO
;  1778          3314  3     BEGIN
;  1779          3315  3     GLOBAL REGISTER
;  1780          3316  3         COUNT=          6,              ! Retrieval pointer count
;  1781          3317  3         LBN=            7,              ! Retrieval pointer LBN
;  1782          3318  3         MAP_POINTER=    8: REF BBLOCK;  ! Pointer to scan map area
;  1783          3319  3     LOCAL
;  1784          3320  3         END_MAP;                        ! Pointer to end of used map area
;  1785          3321  3
;  1786          3322  3
;  1787          3323  3     ! Get pointers to the map area and the end of the used portion
;  1788          3324  3     ! of the map area.
;  1789          3325  3     !
;  1790          3326  3     IF .HEADER[FH2$B_STRUCLEV] EQL 2
;  1791          3327  3     THEN
;  1792          3328  4         BEGIN
;  1793          3329  4         MAP_POINTER = .HEADER + .HEADER[FH2$B_MPOFFSET]*2;
;  1794          3330  4         END_MAP = .MAP_POINTER + .HEADER[FH2$B_MAP_INUSE]*2;
;  1795          3331  4         END
;  1796          3332  3     ELSE
;  1797          3333  4         BEGIN
;  1798          3334  4         MAP_POINTER = .HEADER + .HEADER[FH1$B_MPOFFSET]*2;
;  1799          3335  4         END_MAP = .MAP_POINTER + FM1$C_POINTERS + .MAP_POINTER[FM1$B_INUSE]*2;
;  1800          3336  4         MAP_POINTER = .MAP_POINTER + FM1$C_POINTERS;
;  1801          3337  3         END;
;  1802          3338  3
;  1803          3339  3
;  1804          3340  3     ! Loop until entire map processed.
;  1805          3341  3     !
;  1806          3342  3     UNTIL .MAP_POINTER GEQA .END_MAP DO
;  1807          3343  4         BEGIN
```

```
;  1808          3344  4
;  1809          3345  4          ! Get count and LBN.
;  1810          3346  4          !
;  1811          3347  4          IF .HEADER[FH2$B_STRUCLEV] EQL 2
;  1812          3348  4          THEN
;  1813          3349  4              GET_MAP_POINTER()
;  1814          3350  4          ELSE
;  1815          3351  5              BEGIN
;  1816          3352  5              LBN = .MAP_POINTER[FM1$W_LOWLBN];
;  1817          3353  5              LBN<16,8> = .MAP_POINTER[FM1$B_HIGHLBN];
;  1818          3354  5              COUNT = .MAP_POINTER[FM1$B_COUNT] + 1;
;  1819          3355  5              MAP_POINTER = .MAP_POINTER + 4;
;  1820          3356  4              END;
;  1821          3357  4
;  1822          3358  4
;  1823          3359  4          ! Count into total space.
;  1824          3360  4          !
;  1825          3361  4          CURRENT_MTL[MTL_FILESIZE] = .CURRENT_MTL[MTL_FILESIZE] + .COUNT;
;  1826          3362  4
;  1827          3363  4
;  1828          3364  4          ! Collapse with previous map pointer if contiguous with it and it is on
;  1829          3365  4          ! same RVN -- otherwise, generate new map pointer.
;  1830          3366  4          !
;  1831          3367  4          IF
;  1832          3368  5              BEGIN
;  1833          3369  5              IF .WINDOW[WCB_RVN] NEQ .RVN
;  1834          3370  5              THEN
;  1835          3371  5                  FALSE
;  1836          3372  5              ELSE IF .WINDOW[WCB_SIZE] NEQ 0
;  1837          3373  5              THEN
;  1838          3374  5                  .P[WCB_COUNT] + .P[WCB_LBN] EQL .LBN
;  1839          3375  5              ELSE
;  1840          3376  5                  FALSE
;  1841          3377  5              END
;  1842          3378  4          THEN
;  1843          3379  4              P[WCB_COUNT] = .P[WCB_COUNT] + .COUNT
;  1844          3380  4          ELSE
;  1845          3381  5              BEGIN
;  1846          3382  5              IF .WINDOW[WCB_SIZE] GEQU 255 OR .WINDOW[WCB_RVN] NEQ .RVN
;  1847          3383  5              THEN
;  1848          3384  6                  BEGIN
;  1849          3385  6
;  1850          3386  6                  ! Window block has overflowed.  Move local window block to
;  1851          3387  6                  ! dynamic space and initialize for new block.
;  1852          3388  6                  !
;  1853          3389  6                  WINDOW[WCB_FREE] = 0;
;  1854          3390  6                  DYNWCB = GET_VM(WCB_S_HEADER + .WINDOW[WCB_SIZE]*WCB_S_ENTRY);
;  1855          3391  6                  CH$MOVE(WCB_S_HEADER + .WINDOW[WCB_SIZE] * WCB_S_ENTRY, WINDOW, .DYNWCB);
;  1856          3392  6                  IF .LAST_WINDOW NEQ 0
;  1857          3393  6                      THEN LAST_WINDOW[WCB_LINK] = .DYNWCB;
;  1858          3394  6                  IF ..P_WINDOW EQL 0 THEN .P_WINDOW = .DYNWCB;
;  1859          3395  6                  LAST_WINDOW = .DYNWCB;
;  1860          3396  6                  WINDOW[WCB_LINK] = 0;
;  1861          3397  6                  WINDOW[WCB_VBN] = 0;
;  1862          3398  6                  WINDOW[WCB_SIZE] = 0;
;  1863          3399  6                  WINDOW[WCB_RVN] = .RVN;
;  1864          3400  6                  WINDOW[WCB_FLAGS] = 0;
```

```
; 1865          3401  6                 P = WINDOW + WCB_S_HEADER - WCB_S_ENTRY;
; 1866          3402  5                 END;
; 1867          3403  5
; 1868          3404  5
; 1869          3405  5             ! Generate new pointer.
; 1870          3406  5             !
; 1871          3407  5             WINDOW[WCB_SIZE] = .WINDOW[WCB_SIZE] + 1;
; 1872          3408  5             P = .P + WCB_S_ENTRY;
; 1873          3409  5             P[WCB_COUNT] = .COUNT;
; 1874          3410  5             P[WCB_LBN] = .LBN;
; 1875          3411  4             END;
; 1876          3412  3         END;
; 1877          3413  3
; 1878          3414  3
; 1879          3415  3     ! Allocate a new window at each header boundary. This is necessary
; 1880          3416  3     ! in creating the window for a multi-header ODS-1 index file, since
; 1881          3417  3     ! the call below to read the extension header needs the first part
; 1882          3418  3     ! of the window in place. The de-optimization otherwise caused
; 1883          3419  3     ! is minimal.
; 1884          3420  3     !
; 1885          3421  3     W_SIZE = MAXU(.WINDOW[WCB_SIZE], .WINDOW_SIZE);
; 1886          3422  3     DYNWCB = GET_VM(WCB_S_HEADER + .W_SIZE * WCB_S_ENTRY);
; 1887          3423  3     WINDOW[WCB_FREE] = .W_SIZE - .WINDOW[WCB_SIZE];
; 1888          3424  3     CH$MOVE(WCB_S_HEADER + .WINDOW[WCB_SIZE] * WCB_S_ENTRY, WINDOW, .DYNWCB);
; 1889          3425  3     IF .LAST_WINDOW NEQ 0
; 1890          3426  3         THEN LAST_WINDOW[WCB_LINK] = .DYNWCB;
; 1891          3427  3     IF ..P_WINDOW EQL 0 THEN .P_WINDOW = .DYNWCB;
; 1892          3428  3     LAST_WINDOW = .DYNWCB;
; 1893          3429  3     WINDOW[WCB_VBN] = 0;
; 1894          3430  3     WINDOW[WCB_SIZE] = 0;
; 1895          3431  3     WINDOW[WCB_FLAGS] = 0;
; 1896          3432  3     P = WINDOW + WCB_S_HEADER - WCB_S_ENTRY;
; 1897          3433  3
; 1898          3434  3
; 1899          3435  3     ! If no extension header exists, finish up.
; 1900          3436  3     !
; 1901          3437  3     IF .CURRENT_MTL[MTL_SEQ_DISK]
; 1902          3438  3     OR
; 1903          3439  4         BEGIN
; 1904          3440  4         IF .HEADER[FH2$B_STRUCLEV] EQL 2
; 1905          3441  4         THEN
; 1906          3442  4             .HEADER[FH2$W_EX_FIDNUM] EQL 0
; 1907          3443  4             AND .HEADER[FH2$W_EX_FIDRVN] EQL 0
; 1908          3444  4         ELSE
; 1909          3445  5             BEGIN
; 1910          3446  5             MAP_POINTER = .HEADER + .HEADER[FH2$B_MPOFFSET]*2;
; 1911          3447  5             .MAP_POINTER[FM1$W_EX_FILNUM] EQL 0
; 1912          3448  5             END
; 1913          3449  4         END
; 1914          3450  3     THEN
; 1915          3451  3         EXITLOOP;
; 1916          3452  3
; 1917          3453  3
; 1918          3454  3     ! Get clean file number and RVN.
; 1919          3455  3     !
; 1920          3456  3     IF .HEADER[FH2$B_STRUCLEV] EQL 2
; 1921          3457  3     THEN
```

```
;  1922          3458  4          BEGIN
;  1923          3459  4          EXT_FILE_ID[FID$W_NUM] = .HEADER[FH2$W_EX_FIDNUM];
;  1924          3460  4          EXT_FILE_ID[FID$W_SEQ] = .HEADER[FH2$W_EX_FIDSEQ];
;  1925          3461  4          EXT_FILE_ID[FID$W_RVN] = .HEADER[FH2$W_EX_FIDRVN];
;  1926          3462  4          END
;  1927          3463  3      ELSE
;  1928          3464  4          BEGIN
;  1929          3465  4          EXT_FILE_ID[FID$W_NUM] = .MAP_POINTER[FM1$W_EX_FILNUM];
;  1930          3466  4          EXT_FILE_ID[FID$W_SEQ] = .MAP_POINTER[FM1$W_EX_FILSEQ];
;  1931          3467  4          EXT_FILE_ID[FID$W_RVN] = 1;
;  1932          3468  3          END;
;  1933          3469  3      IF .EXT_FILE_ID[FID$B_RVN] EQL 0 THEN EXT_FILE_ID[FID$B_RVN] = .RVN;
;  1934          3470  3
;  1935          3471  3
;  1936          3472  3      ! Set up header and RVN for next trip through loop.
;  1937          3473  3      !
;  1938          3474  3      HEADER = LOCAL_HEADER;
;  1939          3475  3      RVN = .EXT_FILE_ID[FID$B_RVN];
;  1940          3476  3
;  1941          3477  3
;  1942          3478  3      ! Read extension file header.  If this fails,
;  1943          3479  3      ! exit the loop.
;  1944          3480  3      !
;  1945          3481  3      STATUS = READ_HEADER(EXT_FILE_ID, .HEADER);
;  1946          3482  3      IF NOT .STATUS
;  1947          3483  3      THEN
;  1948          3484  4          BEGIN
;  1949          3485  4          DELETE_WINDOW(..P_WINDOW);
;  1950          3486  4          RETURN .STATUS;
;  1951          3487  3          END;
;  1952          3488  3      WINDOW[WCB_RVN] = .RVN;
;  1953          3489  2      END;
;  1954          3490  2
;  1955          3491  2
;  1956          3492  2 ! Return success.
;  1957          3493  2 !
;  1958          3494  2 SS$_NORMAL
;  1959          3495  1 END;


                                                        .EXTRN  GET_MAP_POINTER

                                        0FFC 00000      .ENTRY  CREATE_WINDOW, Save R2,R3,R4,R5,R6,R7,R8,-       ; 3231
                                                                R9,R10,R11                                       ;
                                5E      F5D8   CE  9E 00002     MOVAB   -2600(SP), SP                            ;
                                          0C   BC  D4 00007     CLRL    @P_WINDOW                                ; 3284
                                50 00000000'   EF  D0 0000A     MOVL    CURRENT_MTL, R0                          ; 3285
                                          20   A0  D4 00011     CLRL    32(R0)                                   ;
                                5B        04   AC  D0 00014     MOVL    P_HEADER, HEADER                         ; 3286
                                          08   AC  DD 00018     PUSHL   P_RVN                                    ; 3287
                                               7E  D4 0001B     CLRL    LAST_WINDOW                              ; 3288
                                          1C   AE  D4 0001D     CLRL    WINDOW                                   ; 3289
                              20  AE      10   AC  D0 00020     MOVL    START_VBN, WINDOW+4                      ; 3290
                                          24   AE  94 00025     CLRB    WINDOW+8                                 ; 3291
                              26  AE      04   AE  9B 00028     MOVZBW  RVN, WINDOW+10                           ; 3292
                                  59      28   AE  9E 0002D     MOVAB   WINDOW+12, P                             ; 3294
```

```
                                02      07   AB   91 00031          CMPB    7(HEADER), #2                           ; 3295
                                        1B   12   00035          BNEQ    1$
                                04      AB   B5   00037          TSTW    4(HEADER)                               ; 3296
                                        16   12   0003A          BNEQ    1$
                                28           6B   91   0003C          CMPB    (HEADER), #40                           ; 3297
                                             11   1F   0003F          BLSSU   1$
                                4C      AB   D5   00041          TSTL    76(HEADER)                              ; 3298
                                        0C   13   00044          BEQL    1$
                        28      AE      4C   AB   D0   00046          MOVL    76(HEADER), WINDOW+12                   ; 3301
                        2C      AE      4C   AB   D0   0004B          MOVL    76(HEADER), WINDOW+16                   ; 3302
                                        08   11   00050          BRB     2$                                      ; 3295
                        28      AE           01   CE   00052  1$:     MNEGL   #1, WINDOW+12                          ; 3306
                        2C      AE           01   CE   00056          MNEGL   #1, WINDOW+16                          ; 3307
                                50      01   AB   9E   0005A  2$:     MOVAB   1(HEADER), R0                          ; 3329
                                        14   AE   D4   0005E          CLRL    20(SP)                                  ; 3326
                                02      07   AB   91   00061          CMPB    7(HEADER), #2
                                        1A   12   00065          BNEQ    3$
                                        14   AE   D6   00067          INCL    20(SP)
                                50           60   9A   0006A          MOVZBL  (R0), R0                                ; 3329
                        0C      AE        6B40   3E   0006D          MOVAW   (HEADER)[R0], 12(SP)
                                58      0C   AE   D0   00072          MOVL    12(SP), MAP_POINTER
                                50      3A   AB   9A   00076          MOVZBL  58(HEADER), R0                          ; 3330
                        10      AE        6840   3E   0007A          MOVAW   (MAP_POINTER)[R0], END_MAP
                                             19   11   0007F          BRB     4$                                      ; 3326
                                50           60   9A   00081  3$:     MOVZBL  (R0), R0                                ; 3334
                        0C      AE        6B40   3E   00084          MOVAW   (HEADER)[R0], 12(SP)
                                58      0C   AE   D0   00089          MOVL    12(SP), MAP_POINTER
                                50      08   A8   9A   0008D          MOVZBL  8(MAP_POINTER), R0                      ; 3335
                        10      AE      0A A840   3E   00091          MOVAW   10(MAP_POINTER)[R0], END_MAP
                                58           0A   C0   00097          ADDL2   #10, MAP_POINTER                        ; 3336
                        10      AE           58   D1   0009A  4$:     CMPL    MAP_POINTER, END_MAP                    ; 3342
                                             03   1F   0009E          BLSSU   5$
                                          00A4   31   000A0          BRW     15$
                                08        14   AE   E9   000A3  5$:     BLBC    20(SP), 6$                              ; 3347
                                   00000000G      00   16   000A7          JSB     GET_MAP_POINTER                         ; 3349
                                             0F   11   000AD          BRB     7$
                                57        02   A8   3C   000AF  6$:     MOVZWL  2(MAP_POINTER), LBN                     ; 3352
                        57              08   10        88   F0   000B3          INSV    (MAP_POINTER)+, #16, #8, LBN            ; 3353
                                56        FD   A8   9A   000B8          MOVZBL  -3(MAP_POINTER), COUNT                  ; 3354
                                56           D6   000BC          INCL    COUNT
                                50 00000000'      EF   D0   000BE  7$:     MOVL    CURRENT_MTL, R0                         ; 3361
                                20      A0        56   C0   000C5          ADDL2   COUNT, 32(R0)
                                             51   D4   000C9          CLRL    R1                                      ; 3369
04      AE      26      AE              08        00   ED   000CB          CMPZV   #0, #8, WINDOW+10, RVN
                                             04   13   000D2          BEQL    8$
                                             51   D6   000D4          INCL    R1
                                             14   11   000D6          BRB     10$
                                        24   AE   95   000D8  8$:     TSTB    WINDOW+8                                ; 3372
                                             0F   13   000DB          BEQL    10$
                        50              69        04   A9   C1   000DD          ADDL3   4(P), (P), R0                        ; 3374
                                57           50   D1   000E2          CMPL    R0, LBN
                                             05   12   000E5          BNEQ    10$
                                69           56   C0   000E7          ADDL2   COUNT, (P)                              ; 3379
                                             AE   11   000EA  9$:     BRB     4$
                                FF      8F   24   AE   91   000EC  10$:    CMPB    WINDOW+8, #255                          ; 3382
                                             03   1E   000F1          BGEQU   11$
                                46           51   E9   000F3          BLBC    R1, 14$
```

```
                                        25  AE  94 000F6 11$:    CLRB    WINDOW+9                                ; 3389
                                    52  24  AE  9A 000F9        MOVZBL  WINDOW+8, R2                            ; 3390
                                    52      08  C4 000FD        MULL2   #8, R2                                  ;
                                    52      14  C0 00100        ADDL2   #20, R2                                 ;
                                            52  DD 00103        PUSHL   R2                                      ;
                            00000000G   00  01  FB 00105        CALLS   #1, GET_VM                              ;
                                  08    AE  50  D0 0010C        MOVL    R0, DYNWCB                              ;
                08  BE          1C    AE  52  28 00110        MOVC3   R2, WINDOW, @DYNWCB                     ; 3391
                                        6E  D5 00116        TSTL    LAST_WINDOW                             ; 3392
                                        05  13 00118        BEQL    12$                                     ;
                                  00    BE      08  AE  D0 0011A        MOVL    DYNWCB, @LAST_WINDOW            ; 3393
                                        0C  BC  D5 0011F 12$:   TSTL    @P_WINDOW                       ; 3394
                                        05  12 00122        BNEQ    13$                                     ;
                                  0C    BC      08  AE  D0 00124        MOVL    DYNWCB, @P_WINDOW               ;
                                  6E      08  AE  D0 00129 13$:   MOVL    DYNWCB, LAST_WINDOW             ; 3395
                                        1C  AE  7C 0012D        CLRQ    WINDOW                          ; 3396
                                        24  AE  94 00130        CLRB    WINDOW+8                        ; 3398
                                  26    AE  04  AE  9B 00133        MOVZBW  RVN, WINDOW+10                  ; 3399
                                  59      28  AE  9E 00138        MOVAB   WINDOW+12, P                    ; 3401
                                        24  AE  96 0013C 14$:   INCB    WINDOW+8                        ; 3407
                                    59      08  C0 0013F        ADDL2   #8, P                           ; 3408
                                    69      56  7D 00142        MOVQ    COUNT, (P)                      ; 3409
                                            A3  11 00145        BRB     9$                              ; 3342
                                    50  24  AE  9A 00147 15$:   MOVZBL  WINDOW+8, R0                    ; 3421
                                  14    AC  50  D1 0014B        CMPL    R0, WINDOW_SIZE                 ;
                                            04  1E 0014F        BGEQU   16$                             ;
                                    50  14  AC  D0 00151        MOVL    WINDOW_SIZE, R0                 ;
                                    5A      50  D0 00155 16$:   MOVL    R0, W_SIZE                      ;
                        7E          5A      03  78 00158        ASHL    #3, W_SIZE, -(SP)               ; 3422
                                    6E      14  C0 0015C        ADDL2   #20, (SP)                       ;
                            00000000G   00  01  FB 0015F        CALLS   #1, GET_VM                      ;
                                  08    AE  50  D0 00166        MOVL    R0, DYNWCB                      ;
                25  AE              5A      24  AE  83 0016A        SUBB3   WINDOW+8, W_SIZE, WINDOW+9      ; 3423
                                    50      24  AE  9A 00170        MOVZBL  WINDOW+8, R0                    ; 3424
                                    50      08  C4 00174        MULL2   #8, R0                          ;
                                    50      14  C0 00177        ADDL2   #20, R0                         ;
                08  BE          1C    AE      50  28 0017A        MOVC3   R0, WINDOW, @DYNWCB             ;
                                        6E  D5 00180        TSTL    LAST_WINDOW                             ; 3425
                                        05  13 00182        BEQL    17$                                     ;
                                  00    BE      08  AE  D0 00184        MOVL    DYNWCB, @LAST_WINDOW            ; 3426
                                        0C  BC  D5 00189 17$:   TSTL    @P_WINDOW                       ; 3427
                                        05  12 0018C        BNEQ    18$                                     ;
                                  0C    BC      08  AE  D0 0018E        MOVL    DYNWCB, @P_WINDOW               ;
                                  6E      08  AE  D0 00193 18$:   MOVL    DYNWCB, LAST_WINDOW             ; 3428
                                        20  AE  D4 00197        CLRL    WINDOW+4                        ; 3429
                                        24  AE  94 0019A        CLRB    WINDOW+8                        ; 3430
                                        27  AE  94 0019D        CLRB    WINDOW+11                       ; 3431
                                    59      28  AE  9E 001A0        MOVAB   WINDOW+12, P                    ; 3432
                                    50  00000000'   EF  D0 001A4        MOVL    CURRENT_MTL, R0         ; 3437
                                    6B      31  A0  E8 001AB        BLBS    49(R0), 26$                     ;
                                    0A      14  AE  E9 001AF        BLBC    20(SP), 19$                     ; 3440
                                            0E  AB  B5 001B3        TSTW    14(HEADER)                      ; 3442
                                            0E  12 001B6        BNEQ    21$                             ;
                                            12  AB  B5 001B8        TSTW    18(HEADER)                      ; 3443
                                            07  11 001BB        BRB     20$                             ;
                                    58      0C  AE  D0 001BD 19$:   MOVL    12(SP), MAP_POINTER             ; 3446
                                            02  A8  B5 001C1        TSTW    2(MAP_POINTER)                  ; 3447
```

```
                                    54  13 001C4 20$:    BEQL    26$                                     ;
                              0C    14  AE  E9 001C6 21$:    BLBC    20(SP), 22$                             ; 3456
                          F8  AD    0E  AB  D0 001CA         MOVL    14(HEADER), EXT_FILE_ID                 ; 3459
                          FC  AD    12  AB  B0 001CF         MOVW    18(HEADER), EXT_FILE_ID+4               ; 3461
                                    09  11 001D4           BRB     23$                                     ; 3456
                          F8  AD    02  A8  D0 001D6 22$:    MOVL    2(MAP_POINTER), EXT_FILE_ID             ; 3465
                          FC  AD        01  B0 001DB         MOVW    #1, EXT_FILE_ID+4                       ; 3467
                                    FC  AD  95 001DF 23$:    TSTB    EXT_FILE_ID+4                           ; 3469
                                    05  12 001E2           BNEQ    24$                                     ;
                          FC  AD    04  AE  90 001E4         MOVB    RVN, EXT_FILE_ID+4                      ;
                              5B  FDF8  CD  9E 001E9 24$:    MOVAB   LOCAL_HEADER, HEADER                    ; 3474
                          04  AE    FC  AD  9A 001EE         MOVZBL  EXT_FILE_ID+4, RVN                      ; 3475
                                    5B  DD 001F3           PUSHL   HEADER                                  ; 3481
                                    F8  AD  9F 001F5         PUSHAB  EXT_FILE_ID                             ;
                        F8E5  CF        02  FB 001F8         CALLS   #2, READ_HEADER                         ;
                          18  AE        50  D0 001FD         MOVL    R0, STATUS                              ;
                              0D    18  AE  E8 00201         BLBS    STATUS, 25$                             ; 3482
                                    0C  BC  DD 00205         PUSHL   @P_WINDOW                               ; 3485
                        0000V   CF        01  FB 00208         CALLS   #1, DELETE_WINDOW                       ;
                              50    18  AE  D0 0020D         MOVL    STATUS, R0                              ; 3486
                                        04 00211           RET                                             ;
                          26  AE    04  AE  90 00212 25$:    MOVB    RVN, WINDOW+10                          ; 3488
                                  FE40  31 00217           BRW     2$                                      ; 3313
                              50        01  D0 0021A 26$:    MOVL    #1, R0                                  ; 3495
                                        04 0021D           RET                                             ;

; Routine Size:  542 bytes,    Routine Base:  CODE + 07CE
```

```
;1961           3496  1 %SBTTL 'DELETE_WINDOW - delete a window block'
;1962           3497  1 ROUTINE DELETE_WINDOW (WINDOW): NOVALUE=
;1963           3498  1
;1964           3499  1 !++
;1965           3500  1 !
;1966           3501  1 ! FUNCTIONAL DESCRIPTION:
;1967           3502  1 !       This routine deletes a window block (or blocks).
;1968           3503  1 !
;1969           3504  1 ! INPUT PARAMETERS:
;1970           3505  1 !       WINDOW          - Pointer to window block.
;1971           3506  1 !
;1972           3507  1 ! IMPLICIT INPUTS:
;1973           3508  1 !       NONE
;1974           3509  1 !
;1975           3510  1 ! OUTPUT PARAMETERS:
;1976           3511  1 !       NONE
;1977           3512  1 !
;1978           3513  1 ! IMPLICIT OUTPUTS:
;1979           3514  1 !       NONE
;1980           3515  1 !
;1981           3516  1 ! ROUTINE VALUE:
;1982           3517  1 !       NONE
;1983           3518  1 !
;1984           3519  1 ! SIDE EFFECTS:
;1985           3520  1 !       Window blocks released.
;1986           3521  1 !
;1987           3522  1 !--
;1988           3523  1
;1989           3524  2 BEGIN
;1990           3525  2 MAP
;1991           3526  2         WINDOW:         REF BBLOCK;     ! Pointer to window block
;1992           3527  2 LOCAL
;1993           3528  2         W:              REF BBLOCK;     ! Pointer to window block
;1994           3529  2
;1995           3530  2
;1996           3531  2 W = .WINDOW;
;1997           3532  2 WHILE .W NEQ 0 DO
;1998           3533  3     BEGIN
;1999           3534  3     LOCAL
;2000           3535  3         NEXT:           REF BBLOCK;     ! Pointer to next window block
;2001           3536  3
;2002           3537  3     NEXT = .W[WCB_LINK];                ! Point to next block
;2003           3538  3     FREE_VM(                            ! Free current block
;2004           3539  3         WCB_S_HEADER + (.W[WCB_SIZE] + .W[WCB_FREE]) * WCB_S_ENTRY,
;2005           3540  3         .W);
;2006           3541  3     W = .NEXT;                          ! Advance to next block
;2007           3542  2     END;
;2008           3543  1 END;


                                000C 00000 DELETE_WINDOW:
                                           .WORD   Save R2,R3                              ;3497
                     52    04   AC  D0 00002         MOVL    WINDOW, W                   ;3531
                                23  13 00006 1$:     BEQL    2$                          ;3532
```

```
                                53              62  D0 00008        MOVL    (W), NEXT                       ; 3537
                                                52  DD 0000B        PUSHL   W                               ; 3540
                                50      08      A2  9A 0000D        MOVZBL  8(W), R0                        ; 3539
                                51      09      A2  9A 00011        MOVZBL  9(W), R1                        ;
                                50              51  C0 00015        ADDL2   R1, R0                          ;
                          7E    50              03  78 00018        ASHL    #3, R0, -(SP)                   ;
                                6E              14  C0 0001C        ADDL2   #20, (SP)                       ;
                    00000000G   00              02  FB 0001F        CALLS   #2, FREE_VM                     ;
                                52              53  D0 00026        MOVL    NEXT, W                         ; 3541
                                                DB  11 00029        BRB     1$                              ; 3532
                                                    04 0002B 2$:    RET                                     ; 3543

; Routine Size:  44 bytes,    Routine Base:  CODE + 09EC
```

```
;2010     3544  1 %SBTTL 'ADD_BLACKHOLE_MAP - add block hole pointer to window'
;2011     3545  1 ROUTINE ADD_BLACKHOLE_MAP (P_WINDOW,COUNT): NOVALUE=
;2012     3546  1
;2013     3547  1 !++
;2014     3548  1 !
;2015     3549  1 ! FUNCTIONAL DESCRIPTION:
;2016     3550  1 !       This routine adds a black hole pointer to the specified window.
;2017     3551  1 !       Write I/O's that map to a black hole pointer are discarded.
;2018     3552  1 !
;2019     3553  1 ! INPUT PARAMETERS:
;2020     3554  1 !       P_WINDOW        - Pointer to window block.
;2021     3555  1 !       COUNT           - Count of blocks.
;2022     3556  1 !
;2023     3557  1 ! IMPLICIT INPUTS:
;2024     3558  1 !       NONE
;2025     3559  1 !
;2026     3560  1 ! OUTPUT PARAMETERS:
;2027     3561  1 !       NONE
;2028     3562  1 !
;2029     3563  1 ! IMPLICIT OUTPUTS:
;2030     3564  1 !       NONE
;2031     3565  1 !
;2032     3566  1 ! ROUTINE VALUE:
;2033     3567  1 !       NONE
;2034     3568  1 !
;2035     3569  1 ! SIDE EFFECTS:
;2036     3570  1 !       NONE
;2037     3571  1 !
;2038     3572  1 !--
;2039     3573  1
;2040     3574  2 BEGIN
;2041     3575  2 LOCAL
;2042     3576  2         W:              REF BBLOCK,     ! Local pointer to window segment
;2043     3577  2         WINDOW:         REF BBLOCK;     ! Address of new window allocated
;2044     3578  2
;2045     3579  2
;2046     3580  2 ! Find the last window block.
;2047     3581  2 !
;2048     3582  2 W = .P_WINDOW;
;2049     3583  2 UNTIL .W[WCB_LINK] EQL 0 DO W = .W[WCB_LINK];
;2050     3584  2
;2051     3585  2
;2052     3586  2 ! If this window block already contains a pointer but is not a black hole,
;2053     3587  2 ! allocate a new one.  Since there is never a need for more than one mapping
;2054     3588  2 ! pointer, only allocate one.  However, if the window block is empty, transform
;2055     3589  2 ! it into a black hole.
;2056     3590  2 !
;2057     3591  2 IF NOT .W[WCB_BLACKHOLE] AND .W[WCB_SIZE] NEQ 0
;2058     3592  2 THEN
;2059     3593  3     BEGIN
;2060     3594  3     WINDOW = GET_VM(WCB_S_ENTRY + WCB_S_HEADER);
;2061     3595  3     W[WCB_LINK] = .WINDOW;
;2062     3596  3     W = .WINDOW;
;2063     3597  3     W[WCB_LINK] = 0;
;2064     3598  3     W[WCB_VBN] = 0;
;2065     3599  3     W[WCB_RVN] = 0;
;2066     3600  3     W[WCB_FREE] = 1;
```

```
;  2067          3601  3      W[WCB_SIZE] = 0;
;  2068          3602  3      W[WCB_FLAGS] = 0;
;  2069          3603  2      END;
;  2070          3604  2
;  2071          3605  2
;  2072          3606  2  ! If no pointer exists yet, initialize it.  There can be no more than one
;  2073          3607  2  ! pointer in a black hole window block.
;  2074          3608  2  !
;  2075          3609  2  IF .W[WCB_SIZE] EQL 0
;  2076          3610  2  THEN
;  2077          3611  3      BEGIN
;  2078          3612  3      W[WCB_BLACKHOLE] = TRUE;
;  2079          3613  3      W[WCB_SIZE] = 1;
;  2080          3614  3      W[WCB_FREE] = .W[WCB_FREE] - 1;
;  2081          3615  3      BBLOCK[.W + WCB_S_HEADER, WCB_COUNT] = 0;
;  2082          3616  3      BBLOCK[.W + WCB_S_HEADER, WCB_LBN] = 0;
;  2083          3617  2      END;
;  2084          3618  2
;  2085          3619  2
;  2086          3620  2  ! Finally add the blocks to the pointer.
;  2087          3621  2  !
;  2088          3622  2  BBLOCK[.W + WCB_S_HEADER, WCB_COUNT] =
;  2089          3623  2      .BBLOCK[.W + WCB_S_HEADER, WCB_COUNT] + .COUNT;
;  2090          3624  1  END;


                                0004 00000 ADD_BLACKHOLE_MAP:
                                           .WORD   Save R2                                  ; 3545
                52      04  AC  D0 00002           MOVL    P_WINDOW, W                     ; 3582
                            62  D5 00006 1$:        TSTL    (W)                             ; 3583
                            05  13 00008           BEQL    2$                              ;
                52          62  D0 0000A           MOVL    (W), W                          ;
                            F7  11 0000D           BRB     1$                              ;
                20      0B  A2  E8 0000F 2$:       BLBS    11(W), 3$                       ; 3591
                        08  A2  95 00013           TSTB    8(W)                            ;
                            1B  13 00016           BEQL    3$                              ;
                            1C  DD 00018           PUSHL   #28                             ; 3594
        00000000G   00          01  FB 0001A       CALLS   #1, GET_VM                      ;
                62              50  D0 00021       MOVL    WINDOW, (W)                     ; 3595
                52              50  D0 00024       MOVL    WINDOW, W                       ; 3596
                                62  7C 00027       CLRQ    (W)                             ; 3597
            09  A2              01  B0 00029       MOVW    #1, 9(W)                        ; 3600
                        08  A2  94  00 0002D       CLRB    8(W)                            ; 3601
                        0B  A2  94  00 00030       CLRB    11(W)                           ; 3602
                        08  A2  95  00 00033 3$:    TSTB    8(W)                            ; 3609
                                11  12 00036       BNEQ    4$                              ;
            0B  A2              01  88 00038       BISB2   #1, 11(W)                       ; 3612
            08  A2              01  90 0003C       MOVB    #1, 8(W)                        ; 3613
                        09  A2  97  00 00040       DECB    9(W)                            ; 3614
                50      14  A2  9E  00 00043       MOVAB   20(W), R0                       ; 3615
                                60  7C 00047       CLRQ    (R0)                            ;
            14  A2          08  AC  C0 00049 4$:    ADDL2   COUNT, 20(W)                    ; 3623
                                04  0004E           RET                                     ; 3624
```

; Routine Size: 79 bytes,   Routine Base: CODE + 0A18

```
;  2092          3625  1 %SBTTL 'ADD_WINDOW_MAP - add pointers to a window'
;  2093          3626  1 ROUTINE ADD_WINDOW_MAP (P_WINDOW,RVN,COUNT,LBN): NOVALUE=
;  2094          3627  1
;  2095          3628  1 !++
;  2096          3629  1 !
;  2097          3630  1 ! FUNCTIONAL DESCRIPTION:
;  2098          3631  1 !       This routine adds a window pointer mapping the specified blocks
;  2099          3632  1 !       to the specified window.  If the window is full or describes a
;  2100          3633  1 !       different RVN, an extension of the same size is added.
;  2101          3634  1 !
;  2102          3635  1 ! INPUT PARAMETERS:
;  2103          3636  1 !       P_WINDOW        - Pointer to window block.
;  2104          3637  1 !       RVN             - Relative volume number of blocks to be added.
;  2105          3638  1 !       COUNT           - Count of blocks to be added.
;  2106          3639  1 !       LBN             - Starting LBN of blocks.
;  2107          3640  1 !
;  2108          3641  1 ! IMPLICIT INPUTS:
;  2109          3642  1 !       NONE
;  2110          3643  1 !
;  2111          3644  1 ! OUTPUT PARAMETERS:
;  2112          3645  1 !       NONE
;  2113          3646  1 !
;  2114          3647  1 ! IMPLICIT OUTPUTS:
;  2115          3648  1 !       NONE
;  2116          3649  1 !
;  2117          3650  1 ! ROUTINE VALUE:
;  2118          3651  1 !       NONE
;  2119          3652  1 !
;  2120          3653  1 ! SIDE EFFECTS:
;  2121          3654  1 !       NONE
;  2122          3655  1 !
;  2123          3656  1 !--
;  2124          3657  1
;  2125          3658  2 BEGIN
;  2126          3659  2 LOCAL
;  2127          3660  2         W:              REF BBLOCK,     ! Local pointer to window segment
;  2128          3661  2         P:              REF BBLOCK,     ! Pointer to window map entry
;  2129          3662  2         SIZE,                           ! Size of new segment to allocate
;  2130          3663  2         WINDOW:         REF BBLOCK;     ! Address of new window allocated
;  2131          3664  2
;  2132          3665  2
;  2133          3666  2 ! Find the last segment of the current window.
;  2134          3667  2 !
;  2135          3668  2 W = .P_WINDOW;
;  2136          3669  2 UNTIL .W[WCB_LINK] EQL 0 DO W = .W[WCB_LINK];
;  2137          3670  2
;  2138          3671  2
;  2139          3672  2 ! If this segment has been used and if it describes blocks on a different
;  2140          3673  2 ! RVN or if this segment is full, allocate a new one.
;  2141          3674  2 !
;  2142          3675  2 IF
;  2143          3676  2     (.W[WCB_SIZE] NEQ 0 AND .W[WCB_RVN] NEQ .RVN) OR
;  2144          3677  2     .W[WCB_FREE] EQL 0
;  2145          3678  2 THEN
;  2146          3679  3     BEGIN
;  2147          3680  3     SIZE = MAXU(.W[WCB_SIZE], 10);
;  2148          3681  3     WINDOW = GET_VM(.SIZE * WCB_S_ENTRY + WCB_S_HEADER);
```

```
;  2149          3682  3     W[WCB_LINK] = .WINDOW;
;  2150          3683  3     W = .WINDOW;
;  2151          3684  3     W[WCB_LINK] = 0;
;  2152          3685  3     W[WCB_VBN] = 0;
;  2153          3686  3     W[WCB_SIZE] = 0;
;  2154          3687  3     W[WCB_FREE] = .SIZE;
;  2155          3688  3     W[WCB_FLAGS] = 0;
;  2156          3689  2     END;
;  2157          3690  2
;  2158          3691  2
;  2159          3692  2 ! Finally add the pointer.  No attempt is made at agglomeration since we
;  2160          3693  2 ! will never allocate an area that could be contiguous.
;  2161          3694  2 !
;  2162          3695  2 P = .W + WCB_S_HEADER + .W[WCB_SIZE]*WCB_S_ENTRY;
;  2163          3696  2 W[WCB_SIZE] = .W[WCB_SIZE] + 1;
;  2164          3697  2 W[WCB_FREE] = .W[WCB_FREE] - 1;
;  2165          3698  2 W[WCB_RVN] = .RVN;
;  2166          3699  2 P[WCB_COUNT] = .COUNT;
;  2167          3700  2 P[WCB_LBN] = .LBN;
;  2168          3701  1 END;


                                000C 00000 ADD_WINDOW_MAP:
                                           .WORD   Save R2,R3                                      ; 3626
                        52      04   AC  D0 00002          MOVL    P_WINDOW, W                    ; 3668
                                     62  D5 00006 1$:       TSTL    (W)                            ; 3669
                                     05  13 00008          BEQL    2$                             ;
                        52           62  D0 0000A          MOVL    (W), W                         ;
                                     F7  11 0000D          BRB     1$                             ;
                                 08  A2  95 0000F 2$:      TSTB    8(W)                           ; 3676
                                     09  13 00012          BEQL    3$                             ;
                08      AC       0A  A2      08           CMPZV   #0, #8, 10(W), RVN             ;
                                     05  12 0001B          BNEQ    4$                             ;
                                 09  A2  95 0001D 3$:      TSTB    9(W)                           ; 3677
                                     2F  12 00020          BNEQ    6$                             ;
                                 50  08  A2  9A 00022 4$:  MOVZBL  8(W), R0                       ; 3680
                                     0A  50  91 00026      CMPB    R0, #10                        ;
                                     03  1E 00029          BGEQU   5$                             ;
                                 50      0A  D0 0002B      MOVL    #10, R0                        ;
                                 53      50  D0 0002E 5$:  MOVL    R0, SIZE                       ;
                            7E   53      03  78 00031      ASHL    #3, SIZE, -(SP)                ; 3681
                                 6E      14  C0 00035      ADDL2   #20, (SP)                      ;
                     00000000G   00      01  FB 00038      CALLS   #1, GET_VM                     ;
                                 62      50  D0 0003F      MOVL    WINDOW, (W)                    ; 3682
                                 52      50  D0 00042      MOVL    WINDOW, W                      ; 3683
                                         62  7C 00045      CLRQ    (W)                            ; 3684
                                 08  A2  94 00047          CLRB    8(W)                           ; 3686
                             09  A2      53  90 0004A      MOVB    SIZE, 9(W)                     ; 3687
                                 0B  A2  94 0004E          CLRB    11(W)                          ; 3688
                                 50  08  A2  9A 00051 6$:  MOVZBL  8(W), R0                       ; 3695
                                 50  14  A240  7E 00055     MOVAQ   20(W)[R0], P                   ;
                                 08  A2  96 0005A          INCB    8(W)                           ; 3696
                                 09  A2  97 0005D          DECB    9(W)                           ; 3697
                             0A  A2      08  AC  90 00060  MOVB    RVN, 10(W)                     ; 3698
```

```
                                60      0C    AC  7D 00065         MOVQ    COUNT, (P)                          ; 3699
                                                  04 00069         RET                                         ; 3701

; Routine Size:  106 bytes,    Routine Base:  CODE + 0A67
```

```
;  2170          3702  1 %SBTTL 'QIO_AST - I/O completion AST routine'
;  2171          3703  1 ROUTINE QIO_AST (VCB): NOVALUE=
;  2172          3704  1
;  2173          3705  1 !++
;  2174          3706  1 !
;  2175          3707  1 ! FUNCTIONAL DESCRIPTION:
;  2176          3708  1 !       This routine is a completion AST routine for the $QIO service in
;  2177          3709  1 !       routine R_W_VIRTUAL.  It decreases the pending I/O count for the
;  2178          3710  1 !       volume, and if it reaches zero, sets event flag 31.
;  2179          3711  1 !
;  2180          3712  1 ! INPUT PARAMETERS:
;  2181          3713  1 !       VCB             - Pointer to VCB.
;  2182          3714  1 !       Remaining standard AST parameters (not used).
;  2183          3715  1 !
;  2184          3716  1 ! IMPLICIT INPUTS:
;  2185          3717  1 !       NONE
;  2186          3718  1 !
;  2187          3719  1 ! OUTPUT PARAMETERS:
;  2188          3720  1 !       NONE
;  2189          3721  1 !
;  2190          3722  1 ! IMPLICIT OUTPUTS:
;  2191          3723  1 !       NONE
;  2192          3724  1 !
;  2193          3725  1 ! ROUTINE VALUE:
;  2194          3726  1 !       NONE
;  2195          3727  1 !
;  2196          3728  1 ! SIDE EFFECTS:
;  2197          3729  1 !       Pending I/O count in VCB decreased.  EFN 31 may be set.
;  2198          3730  1 !
;  2199          3731  1 !--
;  2200          3732  1
;  2201          3733  2 BEGIN
;  2202          3734  2 MAP
;  2203          3735  2         VCB:            REF BBLOCK;
;  2204          3736  2
;  2205          3737  2
;  2206          3738  2 VCB[VCB_IOCOUNT] = .VCB[VCB_IOCOUNT] - 1;
;  2207          3739  2 IF .VCB[VCB_IOCOUNT] LEQ 0 THEN $SETEF(EFN=31);
;  2208          3740  1 END;


                                                        .EXTRN  SYS$SETEF

                                        0000 00000 QIO_AST:.WORD    Save nothing                            ; 3703
                                50      04   AC  D0 00002          MOVL    VCB, R0                         ; 3738
                                0A      A0   B7 00006          :    DECW    10(R0)                          :
                                        09   14 00009          :    BGTR    1$                              ; 3739
                                        1F   DD 0000B          :    PUSHL   #31                             :
                          00000000G     00       01   FB 0000D          CALLS   #1, SYS$SETEF               :
                                                 04 00014 1$:           RET                                 ; 3740

; Routine Size:  21 bytes,    Routine Base:  CODE + 0AD1
```

```
;  2210          3741  1 %SBTTL 'R_W_VIRTUAL - perform read and write virtual'
;  2211          3742  1 ROUTINE R_W_VIRTUAL (EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3)=
;  2212          3743  1
;  2213          3744  1 !++
;  2214          3745  1 !
;  2215          3746  1 ! FUNCTIONAL DESCRIPTION:
;  2216          3747  1 !       This routine maps a read or write virtual block QIO to the appropriate
;  2217          3748  1 !       read or write logical block QIO's using the current window.
;  2218          3749  1 !
;  2219          3750  1 ! INPUT PARAMETERS:
;  2220          3751  1 !       As for $QIO(W) service.
;  2221          3752  1 !       Assumptions: P2 is a multiple of 512, and P2 < 65536
;  2222          3753  1 !
;  2223          3754  1 ! IMPLICIT INPUTS:
;  2224          3755  1 !       CURRENT_MTL     - Pointer to MTL for selected volume set.
;  2225          3756  1 !       CURRENT_WCB     - Pointer to WCB for current file.
;  2226          3757  1 !
;  2227          3758  1 ! OUTPUT PARAMETERS:
;  2228          3759  1 !       NONE
;  2229          3760  1 !
;  2230          3761  1 ! IMPLICIT OUTPUTS:
;  2231          3762  1 !       NONE
;  2232          3763  1 !
;  2233          3764  1 ! ROUTINE VALUE:
;  2234          3765  1 !       SS$_NORMAL if execution is successful or SS$_ENDOFFILE if the
;  2235          3766  1 !       specified virtual block(s) are not completely within the file.
;  2236          3767  1 !
;  2237          3768  1 ! SIDE EFFECTS:
;  2238          3769  1 !       NONE
;  2239          3770  1 !
;  2240          3771  1 !--
;  2241          3772  1
;  2242          3773  2 BEGIN
;  2243          3774  2
;  2244          3775  2 MAP
;  2245          3776  2         IOSB:           REF VECTOR [,WORD];
;  2246          3777  2
;  2247          3778  2 LOCAL
;  2248          3779  2         CHANNEL,                        ! Channel number
;  2249          3780  2         W:              REF BBLOCK,     ! Pointer to window block
;  2250          3781  2         P:              REF BBLOCK,     ! Pointer to window block entry
;  2251          3782  2         N,                              ! VBN mapped so far
;  2252          3783  2         LFUNC:          BBLOCK [4],     ! Logical function code
;  2253          3784  2         LP1, LP2, LP3;                  ! Local copies of P1, P2, P3
;  2254          3785  2
;  2255          3786  2
;  2256          3787  2 ! Pick up local copies of P1, P2, and P3 so that they can advance as the
;  2257          3788  2 ! transfer is done. Note that LP2 is maintained as a block count instead
;  2258          3789  2 ! of as a byte count. Get the appropriate logical function code.
;  2259          3790  2 !
;  2260          3791  2 LP1 = .P1;
;  2261          3792  2 LP2 = .P2 / 512;
;  2262          3793  2 LP3 = .P3;
;  2263          3794  2 LFUNC = .FUNC - IO$_READVBLK + IO$_READLBLK;
;  2264              5  2
;  2265              9  2
;  2266          3797  2 ! The virtual block number must not be 0, and there must be a window.
```

```
; 2267          3798  2 !
; 2268          3799  2 IF .LP3 EQL 0 OR .CURRENT_WCB EQL 0
; 2269          3800  2 THEN
; 2270          3801  2     RETURN SS$_ENDOFFILE;
; 2271          3802  2
; 2272          3803  2 ! If this is a write, check the file's highwater mark. If the write
; 2273          3804  2 ! starts beyond it, back up the VBN to the highwater mark. The mapping
; 2274          3805  2 ! code below will then execute erase functions until we reach the
; 2275          3806  2 ! start of the write.
; 2276          3807  2 !
; 2277          3808  2
; 2278          3809  2 IF .FUNC<0,6> EQL IO$_WRITEVBLK
; 2279          3810  2 THEN
; 2280          3811  3     BEGIN
; 2281          3812  3     IF .LP3 GTRU .CURRENT_WCB[WCB_CUR_HWM]
; 2282          3813  3     AND NOT .CURRENT_MTL[MTL_NOHWM]
; 2283          3814  3     THEN
; 2284          3815  4         BEGIN
; 2285          3816  4         LFUNC = IO$_WRITELBLK OR IO$M_ERASE;
; 2286          3817  4         LP1 = UPLIT (0);
; 2287          3818  4         LP2 = .LP3 - .CURRENT_WCB[WCB_CUR_HWM];
; 2288          3819  4         LP3 = .CURRENT_WCB[WCB_CUR_HWM];
; 2289          3820  3         END;
; 2290          3821  3     END
; 2291          3822  3
; 2292          3823  3 ! If a read goes past the file's highwater mark (and highwater marking
; 2293          3824  3 ! is enforced), stop it at that point and fill the rest of the buffer
; 2294          3825  3 ! with zeroes.
; 2295          3826  3 !
; 2296          3827  3
; 2297          3828  2 ELSE
; 2298          3829  3     BEGIN
; 2299          3830  3     IF .LP2 + .LP3 GTRU .CURRENT_WCB[WCB_CUR_HWM]
; 2300          3831  3     AND NOT .CURRENT_MTL[MTL_NOHWM]
; 2301          3832  3     THEN
; 2302          3833  4         BEGIN
; 2303          3834  4         LP2 = MAX (.CURRENT_WCB[WCB_CUR_HWM] - .LP3, 0);
; 2304          3835  4         CH$FILL (0, .P2 - .[P2*512, .P1 + .LP2*512);
; 2305          3836  3         END;
; 2306          3837  2     END;
; 2307          3838  2
; 2308          3839  2 ! Loop over the window blocks.
; 2309          3840  2 !
; 2310          3841  2 W = .CURRENT_WCB;
; 2311          3842  2 N = .W[WCB_VBN];
; 2312          3843  2 WHILE .W NEQ 0 DO
; 2313          3844  3     BEGIN
; 2314          3845  3     P = .W + WCB_S_HEADER;
; 2315          3846  3
; 2316          3847  3
; 2317          3848  3     ! Loop over the entries within the window block. Maintain the
; 2318          3849  3     ! byte count in IOSB in case we exit.
; 2319          3850  3     !
; 2320          3851  3     DECR I FROM .W[WCB_SIZE] TO 1 DO
; 2321          3852  4         BEGIN
; 2322          3853  4
; 2323          3854  4         ! Since an erase may end and a write begin in the same map pointer,
```

```
; 2324          3855  4         ! loop on the map pointer until it does no good.
; 2325          3856  4         !
; 2326          3857  4         WHILE TRUE DO
; 232           3858  5             BEGIN
; 2328          3859  5
; 2329          3860  5             ! If this entry maps the first VBN of the transfer, do it.
; 2330          3861  5             !
; 2331          3862  5             IF .LP3 GEQU .N AND .LP3 LSSU .N + .P[WCB_COUNT]
; 2332          3863  5             THEN
; 2333          3864  6                 BEGIN
; 2334          3865  6                 LOCAL
; 2335          3866  6                     STATUS,
; 2336          3867  6                     XP2,
; 2337          3868  6                     YP2,
; 2338          3869  6                     XP3;
; 2339          3870  6
; 2340          3871  6                 ! Compute LBN and byte count for this segment and update the
; 2341          3872  6                 ! highwater mark. Then do the I/O function appropriate to the
; 2342          3873  6                 ! circumstances.
; 2343          3874  6                 !
; 2344          3875  6                 XP3 = .P[WCB_LBN] + .LP3 - .N;
; 2345          3876  6                 XP2 = MINU(.LP2, .P[WCB_COUNT] - .LP3 + .N);
; 2346          3877  6                 CURRENT_WCB[WCB_CUR_HWM] = MAXU (.CURRENT_WCB[WCB_CUR_HWM], .LP3 + .XP2);
; 2347          3878  6                 IF .LP2 EQL .XP2 AND .LP1 EQL .P1
; 2348          3879  6                 THEN
; 2349          3880  7                     BEGIN
; 2350          3881  7                     IF .W[WCB_BLACKHOLE]
; 2351          3882  7                     THEN
; 2352          3883  8                         BEGIN
; 2353          3884  8                         $SETEF(EFN=.EFN);
; 2354          3885  8                         IF .IOSB NEQ 0 THEN IOSB[0] = SS$_NORMAL;
; 2355          3886  8                         RETURN SS$_NORMAL;
; 2356          3887  8                         END
; 2357          3888  7                     ELSE
; 2358          3889  8                         BEGIN
; 2359          3890  8                         LOCAL
; 2360          3891  8                             VCB:        REF BBLOCK;
; 2361          3892  8
; 2362          3893  8                         VCB = .CURRENT_MTL[MTL_VCB(.W[WCB_RVN]-.CURRENT_MTL[MTL_RVN_BASE])];
; 2363          3894  8                         CHANNEL = SWITCH_VOLUME(.W[WCB_RVN]);
; 2364        P 3895  8                         STATUS = $QIO(
; 2365        P 3896  8                             FUNC=.LFUNC,
; 2366        P 3897  8                             CHAN=.CHANNEL,
; 2367        P 3898  8                             IOSB=.IOSB,
; 2368        P 3899  8                             EFN=.EFN,
; 2369        P 3900  8                             ASTADR=QIO_AST,
; 2370        P 3901  8                             ASTPRM=.VCB,
; 2371        P 3902  8                             P1=.LP1,
; 2372        P 3903  8                             P2=.XP2*512,
; 2373          3904  8                             P3=.XP3);
; 2374          3905  8
; 2375          3906  8
; 2376          3907  8                         ! Increment the pending I/O count for this volume.
; 2377          3908  8                         !
; 2378          3909  8                         IF .STATUS THEN VCB[VCB_IOCOUNT] = .VCB[VCB_IOCOUNT] + 1;
; 2379          3910  8                         RETURN .STATUS;
; 2380          3911  8                         END
```

```
;  2381          3912  7                    END
;  2382          3913  6                ELSE
;  2383          3914  7                    BEGIN
;  2384          3915  7                    IF NOT .W[WCB_BLACKHOLE]
;  2385          3916  7                    THEN
;  2386          3917  8                        BEGIN
;  2387          3918  8                        LOCAL
;  2388          3919  8                            L_IOSB:     VECTOR[4,WORD];
;  2389          3920  8
;  2390          3921  8                        CHANNEL = SWITCH_VOLUME(.W[WCB_RVN]);
;  2391          3922  8
;  2392          3923  8                        ! All erases get done here. Since the area to be
;  2393          3924  8                        ! erased is potentially huge, repeat the I/O's
;  2394          3925  8                        ! until the count is run out.
;  2395          3926  8                        !
;  2396          3927  8                        YP2 = .XP2;
;  2397          3928  8                        DO
;  2398          3929  9                            BEGIN
;  2399        P 3930  9                            STATUS = $QIOW(
;  2400        P 3931  9                                FUNC=.LFUNC,
;  2401        P 3932  9                                CHAN=.CHANNEL,
;  2402        P 3933  9                                IOSB=L_IOSB,
;  2403        P 3934  9                                P1=.LPT,
;  2404        P 3935  9                                P2=MINU(.YP2, 127)*512,
;  2405          3936  9                                P3=.XP3);
;  2406          3937  9                            IF .STATUS THEN STATUS = .L_IOSB[0];
;  2407          3938  9                            IF NOT .STATUS
;  2408          3939  9                            THEN RETURN .STATUS;
;  2409          3940  9                            YP2 = .YP2 - 127;
;  2410          3941  9                            XP3 = .XP3 + 127;
;  2411          3942  9                            END
;  2412          3943  8                        UNTIL .YP2 LEQ 0;
;  2413          3944  7                        END;
;  2414          3945  6                    END;
;  2415          3946  6                LP2 = .LP2 - .XP2;
;  2416          3947  6                LP3 = .LP3 + .XP2;
;  2417          3948  6
;  2418          3949  6                ! If we are erasing, check if we have arrived at the start
;  2419          3950  6                ! of the actual transfer. If so, set the parameters up
;  2420          3951  6                ! up for the transfer.
;  2421          3952  6                !
;  2422          3953  6                IF .LFUNC[IO$V_ERASE]
;  2423          3954  6                AND .LP2 LEQ 0
;  2424          3955  6                THEN
;  2425          3956  7                    BEGIN
;  2426          3957  7                    LFUNC = .FUNC - IO$_READVBLK + IO$_READLBLK;
;  2427          3958  7                    LP1 = .P1;
;  2428          3959  7                    LP2 = .P2 / 512;
;  2429          3960  7                    LP3 = .P3;
;  2430          3961  7                    END
;  2431          3962  7
;  2432          3963  7                ! Maintain byte count in IOSB and check for completion.
;  2433          3964  7                !
;  2434          3965  6                ELSE
;  2435          3966  7                    BEGIN
;  2436          3967  7                    LP1 = .LP1 + .XP2*512;
;  2437          3968  7                    IF .IOSB NEQ 0
```

```
;  2438          3969  7                    THEN
;  2439          3970  8                        BEGIN
;  2440          3971  8                        IOSB[0] = SS$_NORMAL;
;  2441          3972  8                        IOSB[1] = .P2 - .LP2*512;
;  2442          3973  7                        END;
;  2443          3974  7                    IF .LP2 LEQ 0
;  2444          3975  7                    THEN RETURN SS$_NORMAL;
;  2445          3976  7                    EXITLOOP;
;  2446          3977  6                    END;
;  2447          3978  6                END
;  2448          3979  5            ELSE
;  2449          3980  5                EXITLOOP;
;  2450          3981  5
;  2451          3982  4                END;                        ! end of loop on map pointer
;  2452          3983  4
;  2453          3984  4
;  2454          3985  4            ! Advance to next entry.
;  2455          3986  4            !
;  2456          3987  4            N = .N + .P[WCB_COUNT];
;  2457          3988  4            P = .P + WCB_S_ENTRY;
;  2458          3989  3            END;
;  2459          3990  3
;  2460          3991  3
;  2461          3992  3        ! Advance to next window block.
;  2462          3993  3        !
;  2463          3994  3        W = .W[WCB_LINK];
;  2464          3995  2        END;
;  2465          3996  2
;  2466          3997  2
;  2467          3998  2 ! There were not enough mapping pointers to advance to the specified virtual
;  2468          3999  2 ! block number.  Therefore, return SS$_ENDOFFILE.
;  2469          4000  2 !
;  2470          4001  2 SS$_ENDOFFILE
;  2471          4002  1 END;


                                        00AE6          .BLKB   2
                                00000000 00AE8 P.AAB:  .LONG   0                                                    ;

                                                       .EXTRN  SYS$QIO, SYS$QIOW

                                         0FFC 00000 R_W_VIRTUAL:
                                                       .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                 ; 3742
                                    5E          20 C2 00002        SUBL2   #32, SP                                  ;
                                    5B       1C AC D0 00005        MOVL    P1, LP1                                  ; 3791
                0C  AE      20  AC 00000200 8F C7 00009        DIVL3   #512, P2, 12(SP)                         ; 3792
                                    57       0C AE D0 00013        MOVL    12(SP), LP2                              ;
                                    56       24 AC D0 00017        MOVL    P3, LP3                                  ; 3793
                08  AE      0C  AC          10 C3 0001B        SUBL3   #16, FUNC, 8(SP)                         ; 3794
                            04  AE       08 AE D0 00021        MOVL    8(SP), LFUNC                             ;
                                                56 D5 00026        TSTL    LP3                                      ; 3799
                                                7E 13 00028        BEQL    5$                                       ;
                                       00000000' EF D5 0002A        TSTL    CURRENT_WCB                              ;
                                                76 13 00030        BEQL    5$                                       ;
                                    50 00000000' EF D0 00032        MOVL    CURRENT_WCB, R0                          ; 3812
          30        0C  AC          06          00 ED 00039        CMPZV   #0, #6, FUNC, #48                        ; 3809
```

```
                                        27  12 0003F          BNEQ    1$                                      ;
                    0C    A0                56  D1 00041          CMPL    LP3, 12(R0)                             ; 3812
                                            54  1B 00045          BLEQU   3$                                      ;
                          51 00000000'      EF  D0 00047          MOVL    CURRENT_MTL, R1                         ; 3813
                48  31    A1                02  E0 0004E          BBS     #2, 49(R1), 3$                          ;
                    04    AE      0420      8F  3C 00053          MOVZWL  #1056, LFUNC                            ; 3816
                    5B      A0              AF  9E 00059          MOVAB   P.AAB, LP1                              ; 3817
                57          56        0C    A0  C3 0005D          SUBL3   12(R0), LP3, LP2                        ; 3818
                            56        0C    A0  D0 00062          MOVL    12(R0), LP3                             ; 3819
                                            33  11 00066          BRB     3$                                      ; 3809
                51          57              56  C1 00068 1$:      ADDL3   LP3, LP2, R1                            ; 3830
                    0C      A0              51  D1 0006C          CMPL    R1, 12(R0)                              ;
                                            29  1B 00070          BLEQU   3$                                      ;
                          51 00000000'      EF  D0 00072          MOVL    CURRENT_MTL, R1                         ; 3831
                1D  31      A1              02  E0 00079          BBS     #2, 49(R1), 3$                          ;
                50  0C      A0              56  C3 0007E          SUBL3   LP3, 12(R0), R0                         ; 3834
                                            02  18 00083          BGEQ    2$                                      ;
                                            50  D4 00085          CLRL    R0                                      ;
                            57              50  D0 00087 2$:      MOVL    R0, LP2                                 ;
                50          57              09  78 0008A          ASHL    #9, LP2, R0                             ; 3835
                51  20      AC              50  C3 0008E          SUBL3   R0, P2, R1                              ;
          51    00          6E              00  2C 00093          MOVC5   #0, (SP), #0, R1, @P1[R0]               ;
                                      1C BC40     00098                                                         ;
                          52 00000000'      EF  D0 0009B 3$:      MOVL    CURRENT_WCB, W                          ; 3841
                            58        04    A2  D0 000A2          MOVL    4(W), N                                 ; 3842
                                            52  D5 000A6 4$:      TSTL    W                                       ; 3843
                                            03  12 000A8 5$:      BNEQ    6$                                      ;
                                        017F  31 000AA          BRW     25$                                     ;
                            55        14    A2  9E 000AD 6$:      MOVAB   20(R2), P                               ; 3845
                            6E        08    A2  9A 000B1          MOVZBL  8(W), (SP)                              ; 3851
              10  AE                  6E      01  C1 000B5          ADDL3   #1, (SP), I                             ;
                                        0165  31 000BA          BRW     24$                                     ;
                            58              56  D1 000BD 7$:      CMPL    LP3, N                                  ; 3862
                                            03  1E 000C0          BGEQU   9$                                      ;
                                        0157  31 000C2 8$:      BRW     23$                                     ;
                50          58              65  C1 000C5 9$:      ADDL3   (P), N, R0                              ;
                            50              56  D1 000C9          CMPL    LP3, R0                                 ;
                                            F4  1E 000CC          BGEQU   8$                                      ;
                51          56        04    A5  C1 000CE          ADDL3   4(P), LP3, R1                           ; 3875
                5A          51              58  C3 000D3          SUBL3   N, R1, XP3                              ;
                51          65              56  C3 000D7          SUBL3   LP3, (P), R1                            ; 3876
                            51              58  C0 000DB          ADDL2   N, R1                                   ;
                            50              57  D0 000DE          MOVL    LP2, R0                                 ;
                            51              50  D1 000E1          CMPL    R0, R1                                  ;
                                            03  1B 000E4          BLEQU   10$                                     ;
                            50              51  D0 000E6          MOVL    R1, R0                                  ;
                            53              50  D0 000E9 10$:     MOVL    R0, XP2                                 ;
                          50 00000000'      EF  D0 000EC          MOVL    CURRENT_WCB, R0                         ; 3877
                54          56              53  C1 000F3          ADDL3   XP2, LP3, R4                            ;
                            51        0C    A0  D0 000F7          MOVL    12(R0), R1                              ;
                            54              51  D1 000FB          CMPL    R1, R4                                  ;
                                            03  1E 000FE          BGEQU   11$                                     ;
                            51              54  D0 00100          MOVL    R4, R1                                  ;
                    0C      A0              51  D0 00103 11$:     MOVL    R1, 12(R0)                              ;
                            53              57  D1 00107          CMPL    LP2, XP2                                ; 3878
                                            71  12 0010A          BNEQ    14$                                     ;
                    1C      AC              5B  D1 0010C          CMPL    LP1, P1                                 ;
```

```
                                    6B  12 00110          BNEQ    14$                                     ;
                              59      10   AC  D0 00112          MOVL    IOSB, R9                        ; 3885
                              14      0B   A2  E9 00116          BLBC    11(W), 13$                      ; 3881
                                      04   AC  DD 0011A          PUSHL   EFN                             ; 3884
                    00000000G 00           01  FB 0011D          CALLS   #1, SYS$SETEF                   ;
                                           59  D5 00124          TSTL    R9                              ; 3885
                                           03  13 00126          BEQL    12$                             ;
                              69           01  B0 00128          MOVW    #1, (R9)                        ;
                                       00EA 31  00012B 12$:       BRW     22$                             ; 3889
                              51 00000000' EF  D0 00012E 13$:     MOVL    CURRENT_MTL, R1                 ; 3893
                              50       0A  A2  9A 00135          MOVZBL  10(W), R0                       ;
                              54       30  A1  9A 00139          MOVZBL  48(R1), R4                      ;
                              50           54  C2 0013D          SUBL2   R4, R0                          ;
                              54       34 A140 D0 00140          MOVL    52(R1)[R0], VCB                 ;
                              7E       0A  A2  9A 00145          MOVZBL  10(W), -(SP)                    ; 3894
                        F507  CF           01  FB 00149          CALLS   #1, SWITCH_VOLUME               ;
                          14  AE           50  D0 0014E          MOVL    R0, CHANNEL                     ;
                                           7E  7C 00152          CLRQ    -(SP)                           ; 3904
                                           7E  D4 00154          CLRL    -(SP)                           ;
                                           5A  DD 00156          PUSHL   XP3                             ;
                    7E                 53      09  78 00158          ASHL    #9, XP2, -(SP)              ;
                                     0810  8F  BB 0015C          PUSHR   #^M<R4,R11>                     ;
                                     FE81  CF  9F 00160          PUSHAB  QIO_AST                         ;
                                           59  DD 00164          PUSHL   R9                              ;
                                       28  AE  DD 00166          PUSHL   LFUNC                           ;
                                       3C  AE  DD 00169          PUSHL   CHANNEL                         ;
                                       04  AC  DD 0016C          PUSHL   EFN                             ;
                    00000000G 00           0C  FB 0016F          CALLS   #12, SYS$QIO                    ;
                              4F           50  E9 00176          BLBC    STATUS, 17$                     ; 3909
                                       0A  A4  B6 00179          INCW    10(VCB)                         ;
                                           04  0017C          RET                                     ; 3889
                              56       0B  A2  E8 0017D 14$:     BLBS    11(W), 18$                      ; 3915
                              7E       0A  A2  9A 00181          MOVZBL  10(W), -(SP)                    ; 3921
                        F4CB  CF           01  FB 00185          CALLS   #1, SWITCH_VOLUME               ;
                          14  AE           50  D0 0018A          MOVL    R0, CHANNEL                     ;
                              59           53  D0 0018E          MOVL    XP2, YP2                        ; 3927
                                           7E  7C 00191 15$:     CLRQ    -(SP)                           ; 3936
                                           7E  D4 00193          CLRL    -(SP)                           ;
                                           5A  DD 00195          PUSHL   XP3                             ;
                              51           59  D0 00197          MOVL    YP2, R1                         ;
                        0000007F  8F       51  D1 0019A          CMPL    R1, #127                        ;
                                           04  1B 001A1          BLEQU   16$                             ;
                              51       7F  8F  9A 001A3          MOVZBL  #127, R1                        ;
                    7E        51           09  78 001A7 16$:     ASHL    #9, R1, -(SP)                   ;
                                           5B  DD 001AB          PUSHL   LP1                             ;
                                           7E  7C 001AD          CLRQ    -(SP)                           ;
                                       38  AE  9F 001AF          PUSHAB  L_IOSB                          ;
                                       28  AE  DD 001B2          PUSHL   LFUNC                           ;
                                       3C  AE  DD 001B5          PUSHL   CHANNEL                         ;
                                           7E  D4 001B8          CLRL    -(SP)                           ;
                    00000000G 00           0C  FB 001BA          CALLS   #12, SYS$QIOW                   ;
                              6D           50  E9 001C1          BLBC    STATUS, 26$                     ; 3937
                              50       18  AE  3C 001C4          MOVZWL  L_IOSB, STATUS                  ;
                              66           50  E9 001C8 17$:     BLBC    STATUS, 26$                     ; 3938
                              59       81  A9  9E 001CB          MOVAB   -127(R9), YP2                   ; 3940
                              5A       7F  AA  9E 001CF          MOVAB   127(R10), XP3                   ; 3941
                                           59  D5 001D3          TSTL    YP2                             ; 3943
```

```
                                        BA  14 001D5             BGTR    15$                                     ;
                                57      53  C2 001D7 18$:        SUBL2   XP2, LP2                                ; 3946
                                56      53  C0 001DA             ADDL2   XP2, LP3                                ; 3947
                        18      05      AE  02 E1 001DD          BBC     #2, LFUNC+1, 20$                        ; 3953
                                        57  D5 001E2             TSTL    LP2                                     ; 3954
                                        14  14 001E4             BGTR    20$                                     ;
                                04      AE      08  AE  D0 001E6 MOVL    8(SP), LFUNC                            ; 3957
                                5B      1C      AC  D0 001EB     MOVL    P1, LP1                                 ; 3958
                                57      0C      AE  D0 001EF     MOVL    12(SP), LP2                             ; 3959
                                56      24      AC  D0 001F3     MOVL    P3, LP3                                 ; 3960
                                            FEC3    31 001F7 19$:    BRW     7$                                  ; 3953
                        53              53      09  78 001FA 20$:    ASHL    #9, R3, R3                          ; 3967
                                5B              53  C0 001FE         ADDL2   R3, LP1                             ;
                                50      10      AC  D0 00201         MOVL    10$B, R0                            ; 3968
                                        0D      13  00205            BEQL    21$                                 ;
                                60              01  B0 00207         MOVW    #1, (R0)                            ; 3971
                        51              57      09  78 0020A         ASHL    #9, LP2, R1                         ; 3972
                02      A0      20      AC      51  A3 0020E         SUBW3   R1, P2, 2(R0)                       ;
                                        57  D5 00214 21$:            TSTL    LP2                                 ; 3974
                                        04  14 00216                 BGTR    23$                                 ;
                                50      01  D0 00218 22$:            MOVL    #1, R0                              ; 3975
                                            04  0021B                RET                                         ;
                                58      85  C0 0021C 23$:            ADDL2   (P)+, N                             ; 3987
                                55      04  C0 0021F                ADDL2   #4, P                               ; 3988
                                D1      10  AE  F5 00222 24$:        SOBGTR  I, 19$                              ; 3851
                                52          62  D0 00226            MOVL    (W), W                              ; 3994
                                        FE7A    31 00229             BRW     4$                                  ; 3843
                                50      0870    8F  3C 0022C 25$:    MOVZWL  #2160, R0                           ; 4002
                                            04  00231 26$:           RET                                         ;

; Routine Size:  562 bytes,    Routine Base:  CODE + 0AEC
```

```
;  2473          4003  1 %SBTTL 'READ_HOMEBLOCK - read volume home block'
;  2474          4004  1 ROUTINE READ_HOMEBLOCK (CHANNEL,DEVICE_CHAR,HOME_BLOCK): NOVALUE=
;  2475          4005  1
;  2476          4006  1 !++
;  2477          4007  1 !
;  2478          4008  1 ! FUNCTIONAL DESCRIPTION:
;  2479          4009  1 !       This routine reads the home block from a specified volume.
;  2480          4010  1 !
;  2481          4011  1 ! INPUT PARAMETERS:
;  2482          4012  1 !       CHANNEL         - Channel number assigned to device.
;  2483          4013  1 !       DEVICE_CHAR     - Device characteristics of device.
;  2484          4014  1 !       HOME_BLOCK      - Buffer into which home block will be read.
;  2485          4015  1 !
;  2486          4016  1 ! IMPLICIT INPUTS:
;  2487          4017  1 !       CURRENT_VCB
;  2488          4018  1 !
;  2489          4019  1 ! OUTPUT PARAMETERS:
;  2490          4020  1 !       NONE
;  2491          4021  1 !
;  2492          4022  1 ! IMPLICIT OUTPUTS:
;  2493          4023  1 !       NONE
;  2494          4024  1 !
;  2495          4025  1 ! ROUTINE VALUE:
;  2496          4026  1 !       NONE
;  2497          4027  1 !
;  2498          4028  1 ! SIDE EFFECTS:
;  2499          4029  1 !       NONE
;  2500          4030  1 !
;  2501          4031  1 !--
;  2502          4032  1
;  2503          4033  2 BEGIN
;  2504          4034  2 MAP
;  2505          4035  2         HOME_BLOCK:     REF BBLOCK,     ! Pointer to home block buffer
;  2506          4036  2         DEVICE_CHAR:    REF BBLOCK;     ! Pointer to device characteristics
;  2507          4037  2 LOCAL
;  2508          4038  2         DELTA,                          ! Home block search delta
;  2509          4039  2         BLOCKFACT,                      ! Device blocking factor
;  2510          4040  2         STATUS,                         ! Status variable
;  2511          4041  2         IOSB:           VECTOR[4,WORD]; ! I/O status block
;  2512          4042  2
;  2513          4043  2
;  2514          4044  2 ! Compute the home block search delta from the volume geometry, according
;  2515          4045  2 ! to the following rules, where volume geometry is expressed in the order
;  2516          4046  2 ! sectors, tracks, cylinders:
;  2517          4047  2 !
;  2518          4048  2 !       n x 1 x 1:      1
;  2519          4049  2 !       1 x n x 1:      1
;  2520          4050  2 !       1 x 1 x n:      1
;  2521          4051  2 !
;  2522          4052  2 !       n x m x 1:      n+1
;  2523          4053  2 !       n x 1 x m:      n+1
;  2524          4054  2 !       1 x n x m:      n+1
;  2525          4055  2 !
;  2526          4056  2 !       s x t x c:      (t+1)*s+1
;  2527          4057  2 !
;  2528          4058  3 BLOCKFACT = (.DEVICE_CHAR[DIB$B_SECTORS]
;  2529          4059  3                 * .DEVICE_CHAR[DIB$B_TRACKS]
```

```
;  2530          4060  3                 * .DEVICE_CHAR[DIB$W_CYLINDERS])
;  2531          4061  2                 / .DEVICE_CHAR[DIB$L_MAXBLOCK];
;  2532          4062  2
;  2533          4063  2
;  2534          4064  2 DELTA = 1;
;  2535          4065  2
;  2536          4066  2
;  2537          4067  2 IF
;  2538          4068  2     .DEVICE_CHAR[DIB$W_CYLINDERS] GTR 1 AND
;  2539          4069  2     .DEVICE_CHAR[DIB$B_TRACKS] GTR 1
;  2540          4070  2 THEN
;  2541          4071  2     DELTA = .DELTA + .DEVICE_CHAR[DIB$B_TRACKS];
;  2542          4072  2
;  2543          4073  2
;  2544          4074  2 IF
;  2545          4075  2     .DEVICE_CHAR[DIB$B_SECTORS] GTR 1 AND
;  2546          4076  3     (.DEVICE_CHAR[DIB$W_CYLINDERS] GTR 1 OR
;  2547          4077  3         .DEVICE_CHAR[DIB$B_TRACKS] GTR 1)
;  2548          4078  2 THEN
;  2549          4079  2     DELTA = (.DELTA * .DEVICE_CHAR[DIB$B_SECTORS] + .BLOCKFACT) / .BLOCKFACT;
;  2550          4080  2
;  2551          4081  2
;  2552          4082  2 IF
;  2553          4083  2     .DELTA EQL 0 OR
;  2554          4084  2     .DELTA GTRU .DEVICE_CHAR[DIB$L_MAXBLOCK] / 10
;  2555          4085  2 THEN
;  2556          4086  2     DELTA = 1;
;  2557          4087  2
;  2558          4088  2
;  2559          4089  2 ! Search for a valid home block.  The loop terminates when one is found.
;  2560          4090  2 ! If an error other than a surface error occurs, or when the entire
;  2561          4091  2 ! volume has been examined and no valid home block found, a fatal error
;  2562          4092  2 ! is signalled.
;  2563          4093  2 !
;  2564          4094  2 INCRU LBN FROM 1 BY .DELTA DO
;  2565          4095  3     BEGIN
;  2566        P 4096  3     STATUS = $QIOW(
;  2567        P 4097  3         FUNC=IO$_READLBLK,
;  2568        P 4098  3         CHAN=.CHANNEL,
;  2569        P 4099  3         IOSB=IOSB,
;  2570        P 4100  3         P1=.HOME_BLOCK,
;  2571        P 4101  3         P2=512,
;  2572          4102  3         P3=.LBN);
;  2573          4103  3     IF .STATUS THEN STATUS = .IOSB[0];
;  2574          4104  3     IF .STATUS
;  2575          4105  3     THEN
;  2576          4106  4         BEGIN
;  2577          4107  4         IF
;  2578          4108  5             BEGIN
;  2579          4109  5             SELECTONE .HOME_BLOCK[HM2$B_STRUCLEV] OF
;  2580          4110  5                 SET
;  2581          4111  5
;  2582          4112  5                 [1]:
;  2583          4113  6                     BEGIN
;  2584          4114  7                     (.HOME_BLOCK[HM1$W_STRUCLEV] EQL HM1$C_LEVEL1 OR
;  2585          4115  6                      .HOME_BLOCK[HM1$W_STRUCLEV] EQL HM1$C_LEVEL2) AND
;  2586          4116  6                     .HOME_BLOCK[HM1$W_IBMAPSIZE] NEQ 0 AND
```

```
; 2587          4117  6                   .HOME_BLOCK[HM1$L_IBMAPLBN] NEQ 0 AND
; 2588          4118  6                   .HOME_BLOCK[HM1$W_MAXFILES] NEQ 0 AND
; 2589          4119  6                   .HOME_BLOCK[HM1$W_CLUSTER] NEQ 0 AND
; 2590          4120  6                   CHECKSUM2(.HOME_BLOCK, $BYTEOFFSET(HM1$W_CHECKSUM1)) AND
; 2591          4121  6                   CHECKSUM2(.HOME_BLOCK, $BYTEOFFSET(HM1$W_CHECKSUM2))
; 2592          4122  5                   END;
; 2593          4123  5
; 2594          4124  5
; 2595          4125  5               [2]:
; 2596          4126  6                   BEGIN
; 2597          4127  6                   .HOME_BLOCK[HM2$L_HOMELBN] EQL .LBN AND
; 2598          4128  6                   .HOME_BLOCK[HM2$L_ALTIDXLBN] NEQ 0 AND
; 2599          4129  6                   .HOME_BLOCK[HM2$W_CLUSTER] NEQ 0 AND
; 2600          4130  6                   .HOME_BLOCK[HM2$W_HOMEVBN] NEQ 0 AND
; 2601          4131  6                   .HOME_BLOCK[HM2$W_ALHOMEVBN] NEQ 0 AND
; 2602          4132  6                   .HOME_BLOCK[HM2$W_ALTIDXVBN] NEQ 0 AND
; 2603          4133  6                   .HOME_BLOCK[HM2$W_IBMAPVBN] NEQ 0 AND
; 2604          4134  6                   .HOME_BLOCK[HM2$L_IBMAPLBN] NEQ 0 AND
; 2605          4135  6                   .HOME_BLOCK[HM2$L_MAXFILES] NEQ 0 AND
; 2606          4136  6                   .HOME_BLOCK[HM2$W_IBMAPSIZE] NEQ 0 AND
; 2607          4137  6                   .HOME_BLOCK[HM2$W_RESFILES] NEQ 0 AND
; 2608          4138  6                   CHECKSUM2(.HOME_BLOCK, $BYTEOFFSET(HM2$W_CHECKSUM1)) AND
; 2609          4139  6                   CHECKSUM2(.HOME_BLOCK, $BYTEOFFSET(HM2$W_CHECKSUM2))
; 2610          4140  5                   END;
; 2611          4141  5
; 2612          4142  5
; 2613          4143  5               [OTHERWISE]:
; 2614          4144  5                   FALSE;
; 2615          4145  5
; 2616          4146  5
; 2617          4147  5                       TES
; 2618          4148  5                   END
; 2619          4149  4               THEN
; 2620          4150  4                   EXITLOOP;
; 2621          4151  4               END
; 2622          4152  3           ELSE
; 2623          4153  4               BEGIN
; 2624          4154  4               IF .STATUS EQL SS$_ILLBLKNUM
; 2625          4155  4               THEN
; 2626          4156  4                   SIGNAL(BACKUP$_NOHOMEBLK, 1, CURRENT_VCB[VCB_DEVICE]);
; 2627          4157  4
; 2628          4158  4               IF
; 2629          4159  4                   .STATUS NEQ SS$_PARITY AND
; 2630          4160  4                   .STATUS NEQ SS$_FORMAT AND
; 2631          4161  4                   .STATUS NEQ SS$_DATACHECK
; 2632          4162  4               THEN
; 2633          4163  4                   SIGNAL(BACKUP$_READERR + STS$K_SEVERE, 1, CURRENT_VCB[VCB_DEVICE], .STATUS);
; 2634          4164  3               END;
; 2635          4165  2           END;
; 2636          4166  1 END;


                                01FC 00000 READ_HOMEBLOCK:
                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8                       ; 4004
```

```
                                58 00000000G   00   9E 00002          MOVAB   LIB$SIGNAL, R8                                  ;
                                57 00000000'   EF   9E 00009          MOVAB   CURRENT_VCB, R7                               ;
                                56 00000000G   00   9E 00010          MOVAB   CHECKSUM2, R6                                 ;
                                5E             08   C2 00017          SUBL2   #8, SP                                        ;
                                50         08  AC   D0 0001A          MOVL    DEVICE_CHAR, R0                               ; 4058
                                51         08  A0   9A 0001E          MOVZBL  8(R0), R1                                     ; 4059
                                52         09  A0   9A 00022          MOVZBL  9(R0), R2                                     ;
                                51             52   C4 00026          MULL2   R2, R1                                        ;
                                53         0A  A0   3C 00029          MOVZWL  10(R0), R3                                    ; 4060
                                51             53   C4 0002D          MULL2   R3, R1                                        ;
                           53   51         70  A0   C7 00030          DIVL3   112(R0), R1, BLOCKFACT                        ; 4061
                                52             01   D0 00035          MOVL    #1, DELTA                                     ; 4064
                                               51   D4 00038          CLRL    R1                                            ; 4068
                                01         0A  A0   B1 0003A          CMPW    10(R0), #1                                    ;
                                               0F   1B 0003E          BLEQU   1$                                            ;
                                               51   D6 00040          INCL    R1                                            ;
                                01         09  A0   91 00042          CMPB    9(R0), #1                                     ; 4069
                                               07   1B 00046          BLEQU   1$                                            ;
                                54         09  A0   9A 00048          MOVZBL  9(R0), R4                                     ; 4071
                                52             54   C0 0004C          ADDL2   R4, DELTA                                     ;
                                01         08  A0   91 0004F 1$:      CMPB    8(R0), #1                                     ; 4075
                                               17   1B 00053          BLEQU   3$                                            ;
                                06             51   E8 00055          BLBS    R1, 2$                                        ; 4076
                                01         09  A0   91 00058          CMPB    9(R0), #1                                     ; 4077
                                               0E   1B 0005C          BLEQU   3$                                            ;
                                51         08  A0   9A 0005E 2$:      MOVZBL  8(R0), R1                                     ; 4079
                                51             52   C4 00062          MULL2   DELTA, R1                                     ;
                                51             53   C0 00065          ADDL2   BLOCKFACT, R1                                 ;
                           52   51             53   C7 00068          DIVL3   BLOCKFACT, R1, DELTA                          ;
                                               52   D5 0006C 3$:      TSTL    DELTA                                         ; 4083
                                               0A   13 0006E          BEQL    4$                                            ;
                      50   70   A0             0A   C7 00070          DIVL3   #10, 112(R0), R0                              ; 4084
                                50             52   D1 00075          CMPL    DELTA, R0                                     ;
                                               03   1B 00078          BLEQU   5$                                            ;
                                52             01   D0 0007A 4$:      MOVL    #1, DELTA                                     ; 4086
                                53         0C  AC   D0 0007D 5$:      MOVL    HOME_BLOCK, R3                                ; 4102
                                55             01   D0 00081          MOVL    #1, LBN                                       ;
                                               7E   7C 00084 6$:      CLRQ    -(SP)                                         ;
                                               7E   D4 00086          CLRL    -(SP)                                         ;
                                               55   DD 00088          PUSHL   LBN                                           ;
                                7E     0200    8F   3C 0008A          MOVZWL  #512, -(SP)                                   ;
                                               53   DD 0008F          PUSHL   R3                                            ;
                                               7E   7C 00091          CLRQ    -(SP)                                         ;
                                           20  AE   9F 00093          PUSHAB  IOSB                                          ;
                                               21   DD 00096          PUSHL   #33                                           ;
                                           04  AC   DD 00098          PUSHL   CHANNEL                                       ;
                                               7E   D4 0009B          CLRL    -(SP)                                         ;
                      00000000G    00             0C   FB 0009D          CALLS   #12, SYS$QIOW                                 ;
                                54             50   D0 000A4          MOVL    R0, STATUS                                    ;
                                03             54   E9 000A7          BLBC    STATUS, 7$                                    ; 4103
                                54             6E   3C 000AA          MOVZWL  IOSB, STATUS                                  ;
                                03             54   E8 000AD 7$:      BLBS    STATUS, 8$                                    ; 4104
                                           0089    31   000B0          BRW     17$                                           ;
                                50         0D  A3   9A 000B3 8$:      MOVZBL  13(R3), R0                                    ; 4109
                                01             50   91 000B7          CMPB    R0, #1                                        ; 4112
                                               29   12 000BA          BNEQ    12$                                           ;
                          0101  8F         0C  A3   B1 000BC          CMPW    12(R3), #257                                  ; 4114
```

```
                                        08  13 000C2          BEQL    9$                                      ;
                                0102    8F      0C  A3  B1 000C4          CMPW    12(R3), #258                            ;4115
                                                    21  12 000CA          BNEQ    13$                                     ;
                                                    63  B5 000CC 9$:       TSTW    (R3)                                    ;4116
                                                    52  13 000CE          BEQL    16$                                     ;
                                                02  A3  D5 000D0          TSTL    2(R3)                                   ;4117
                                                    03  12 000D3          BNEQ    10$                                     ;
                                                0083    31 000D5          BRW     19$                                     ;
                                                06  A3  B5 000D8 10$:     TSTW    6(R3)                                   ;4118
                                                    03  12 000DB          BNEQ    11$                                     ;
                                                0084    31 000DD          BRW     20$                                     ;
                                                08  A3  B5 000E0 11$:     TSTW    8(R3)                                   ;4119
                                                    3D  11 000E3          BRB     16$                                     ;
                                            02          50  91 000E5 12$:     CMPB    R0, #2                                  ;4125
                                                    03  12 000E8          BNEQ    13$                                     ;
                                            55          63  D1 000EA          CMPL    (R3), LBN                               ;4127
                                                    03  13 000ED 13$:     BEQL    15$                                     ;
                                                008E    31 000EF 14$:     BRW     21$                                     ;
                                                08  A3  D5 000F2 15$:     TSTL    8(R3)                                   ;4128
                                                    F8  13 000F5          BEQL    14$                                     ;
                                                0E  A3  B5 000F7          TSTW    14(R3)                                  ;4129
                                                    F3  13 000FA          BEQL    14$                                     ;
                                                10  A3  B5 000FC          TSTW    16(R3)                                  ;4130
                                                    7F  13 000FF          BEQL    21$                                     ;
                                                12  A3  B5 00101          TSTW    18(R3)                                  ;4131
                                                    7A  13 00104          BEQL    21$                                     ;
                                                14  A3  B5 00106          TSTW    20(R3)                                  ;4132
                                                    75  13 00109          BEQL    21$                                     ;
                                                16  A3  B5 0010B          TSTW    22(R3)                                  ;4133
                                                    70  13 0010E          BEQL    21$                                     ;
                                                18  A3  D5 00110          TSTL    24(R3)                                  ;4134
                                                    6B  13 00113          BEQL    21$                                     ;
                                                1C  A3  D5 00115          TSTL    28(R3)                                  ;4135
                                                    66  13 00118          BEQL    21$                                     ;
                                                20  A3  B5 0011A          TSTW    32(R3)                                  ;4136
                                                    61  13 0011D          BEQL    21$                                     ;
                                                22  A3  B5 0011F          TSTW    34(R3)                                  ;4137
                                                    5C  13 00122 16$:     BEQL    21$                                     ;
                                                    3A  DD 00124          PUSHL   #58                                     ;4138
                                                    53  DD 00126          PUSHL   R3
                                            66          02  FB 00128          CALLS   #2, CHECKSUM2
                                            52          50  E9 0012B          BLBC    R0, 21$
                                            7E      01FE    8F  3C 0012E          MOVZWL  #510, -(SP)                     ;4139
                                                    53  DD 00133          PUSHL   R3
                                            66          02  FB 00135          CALLS   #2, CHECKSUM2
                                            45          50  E9 00138          BLBC    R0, 21$
                                                        04  00139          RET
                            000000DC    8F              54  D1 0013C 17$:     CMPL    STATUS, #220                            ;4154
                                                    0F  12 00143          BNEQ    18$                                     ;
                        7E                  67          20  C1 00145          ADDL3   #32, CURRENT_VCB, -(SP)                 ;4156
                                                    01  DD 00149          PUSHL   #1
                                        00000000G       8F  DD 0014B          PUSHL   #BACKUP$_NOHOMEBLK
                                            68          03  FB 00151          CALLS   #3, LIB$SIGNAL                          ;
                            000001F4    8F              54  D1 00154 18$:     CMPL    STATUS, #500                            ;4159
                                                    23  13 0015B 19$:     BEQL    21$                                     ;
                            000000BC    8F              54  D1 0015D          CMPL    STATUS, #188                            ;4160
                                                    1A  13 00164 20$:     BEQL    21$                                     ;
```

```
                          0000005C    8F              54  D1 00166           CMPL    STATUS, #92                             ; 4161
                                                      11  13 0016D           BEQL    21$                                     ;
                                                      54  DD 0016F           PUSHL   STATUS                                  ; 4163
                                7E            67      20  C1 00171           ADDL3   #32, CURRENT_VCB, -(SP)                 ;
                                                      01  DD 00175           PUSHL   #1                                      ;
                                          00000000G   8F  DD 00177           PUSHL   #BACKUP$_READERR+4                      ;
                                              68      04  FB 0017D           CALLS   #4, LIB$SIGNAL                          ;
                                              55      52  C0 00180 21$:      ADDL2   DELTA, LBN                              ; 4094
                                                  FEFE  31  00183           BRW     6$                                      ;

; Routine Size:  390 bytes,    Routine Base:  CODE + 0D1E
```

```
; 2638          4167  1 %SBTTL 'STA_INIVOL - initialize volume'
; 2639          4168  1 GLOBAL ROUTINE STA_INIVOL : NOVALUE=
; 2640          4169  1
; 2641          4170  1 !++
; 2642          4171  1 !
; 2643          4172  1 ! FUNCTIONAL DESCRIPTION:
; 2644          4173  1 !       This routine initializes a volume.
; 2645          4174  1 !
; 2646          4175  1 ! INPUT PARAMETERS:
; 2647          4176  1 !       NONE
; 2648          4177  1 !
; 2649          4178  1 ! IMPLICIT INPUTS:
; 2650          4179  1 !       OUTPUT_ATTBUF   - Contains volume summary attributes.
; 2651          4180  1 !       OUTPUT_MTL      - Pointer to MTL for output volume set.
; 2652          4181  1 !
; 2653          4182  1 ! OUTPUT PARAMETERS:
; 2654          4183  1 !       NONE
; 2655          4184  1 !
; 2656          4185  1 ! IMPLICIT OUTPUTS:
; 2657          4186  1 !       NONE
; 2658          4187  1 !
; 2659          4188  1 ! ROUTINE VALUE:
; 2660          4189  1 !       NONE
; 2661          4190  1 !
; 2662          4191  1 ! SIDE EFFECTS:
; 2663          4192  1 !       NONE
; 2664          4193  1 !
; 2665          4194  1 !--
; 2666          4195  1
; 2667          4196  2 BEGIN
; 2668          4197  2 LOCAL
; 2669          4198  2         HOME_BLOCK:     BBLOCK[512],    ! Home block buffer
; 2670          4199  2         RVN,                            ! Actual RVN
; 2671          4200  2         CHANNEL,                        ! Channel number
; 2672          4201  2         DESC:           VECTOR[2],      ! Descriptor
; 2673          4202  2         DEVCHAR:        BBLOCK[DIB$C_LENGTH],   ! Device characteristics
; 2674          4203  2         STATUS;                         ! Status variable
; 2675          4204  2
; 2676          4205  2
; 2677          4206  2 ! Find the MTL and VCB for the volume to be initialized.
; 2678          4207  2 !
; 2679          4208  2 RVN = .OUTPUT_ATTBUF[VSR_RVN];
; 2680          4209  2 IF .RVN EQL 0 THEN RVN = 1;
; 2681          4210  2 CURRENT_MTL = .OUTPUT_MTL;
; 2682          4211  2
; 2683          4212  2
; 2684          4213  2 ! Make sure the number of output volumes specified in the command matches
; 2685          4214  2 ! the number specified in the save set.
; 2686          4215  2 !
; 2687          4216  2 IF NOT .QUAL[QUAL_VOLU]
; 2688          4217  2 THEN
; 2689          4218  2     IF
; 2690          4219  3         BEGIN
; 2691          4220  3         IF .COM_O_SETCOUNT EQL 0
; 2692          4221  3             THEN .CURRENT_MTL[MTL_SETCOUNT] NEQ 1
; 2693          4222  3             ELSE .CURRENT_MTL[MTL_SETCOUNT] NEQ .COM_O_SETCOUNT
; 2694          4223  3         END
```

```
;  2695     4224  2      THEN
;  2696     4225  2          SIGNAL(BACKUP$_BADSETCNT);
;  2697     4226  2
;  2698     4227  2
;  2699     4228  2  ! Make sure the relative volume number specified in the save set is
;  2700     4229  2  ! within range.
;  2701     4230  2  !
;  2702     4231  2  IF .RVN-.CURRENT_MTL[MTL_RVN_BASE] GTRU .CURRENT_MTL[MTL_SETCOUNT]
;  2703     4232  2  THEN
;  2704     4233  2      SIGNAL(BACKUP$_INVATTVAL);
;  2705     4234  2
;  2706     4235  2
;  2707     4236  2  ! Assign a channel to the volume and get the device characteristics.
;  2708     4237  2  !
;  2709     4238  2  CURRENT_VCB = .CURRENT_MTL[MTL_VCB(.RVN-.CURRENT_MTL[MTL_RVN_BASE])];
;  2710     4239  2  CHANNEL = SWITCH_VOLUME(.RVN);
;  2711     4240  2  DESC[0] = DIB$C_LENGTH;
;  2712     4241  2  DESC[1] = DEVCHAR;
;  2713     4242  2  STATUS = $GETCHN(CHAN=.CHANNEL, PRIBUF=DESC);
;  2714     4243  2  IF NOT .STATUS
;  2715     4244  2  THEN
;  2716     4245  2      SIGNAL(BACKUP$_GETCHN, 1, CURRENT_VCB[VCB_DEVICE]);
;  2717     4246  2
;  2718     4247  2
;  2719     4248  2  ! If the initialization parameters are to be propagated from the output
;  2720     4249  2  ! volume, read its home block and reinitialize OUTPUT_ATTBUF as appropriate.
;  2721     4250  2  !
;  2722     4251  2  IF NOT .QUAL[QUAL_INIT]
;  2723     4252  2  THEN
;  2724     4253  3      BEGIN
;  2725     4254  3
;  2726     4255  3      ! Read the home block.
;  2727     4256  3      !
;  2728     4257  3      READ_HOMEBLOCK(.CHANNEL, DEVCHAR, HOME_BLOCK);
;  2729     4258  3
;  2730     4259  3
;  2731     4260  3      ! Make sure that the new structure level matches the old.  This is the
;  2732     4261  3      ! only initialization parameter that must not change.
;  2733     4262  3      !
;  2734     4263  3      IF .HOME_BLOCK[HM2$B_STRUCLEV] NEQ .(OUTPUT_ATTBUF[VSR_VOLSTRUCT])<8,8>
;  2735     4264  3      THEN
;  2736     4265  3          SIGNAL(BACKUP$_STRUCLEV, 1, CURRENT_VCB[VCB_DEVICE]);
;  2737     4266  3
;  2738     4267  3
;  2739     4268  3      ! Reinitialize the OUTPUT_ATTBUF area with the initialization parameters
;  2740     4269  3      ! from the output volume.
;  2741     4270  3      !
;  2742     4271  3      IF .HOME_BLOCK[HM2$B_STRUCLEV] EQL 2
;  2743     4272  3      THEN
;  2744     4273  4          BEGIN
;  2745     4274  4          BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH] = HM2$S_VOLNAME;
;  2746     4275  4          BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$A_POINTER] = HOME_BLOCK[HM2$T_VOLNAME];
;  2747     4276  4          BBLOCK[OUTPUT_ATTBUF[VSR_OWNERNAME], DSC$W_LENGTH] = HM2$S_OWNERNAME;
;  2748     4277  4          BBLOCK[OUTPUT_ATTBUF[VSR_OWNERNAME], DSC$A_POINTER] = HOME_BLOCK[HM2$T_OWNERNAME];
;  2749     4278  4          (OUTPUT_ATTBUF[VSR_VOLDATE]) = .(HOME_BLOCK[HM2$Q_CREDATE]);
;  2750     4279  4          (OUTPUT_ATTBUF[VSR_VOLDATE]+4) = .(HOME_BLOCK[HM2$Q_CREDATE]+4);
;  2751     4280  4          OUTPUT_ATTBUF[VSR_VOLOWNER] = .HOME_BLOCK[HM2$L_VOLOWNER];
```

```
;  2752          4281  4          OUTPUT_ATTBUF[VSR_MAXFILES] = .HOME_BLOCK[HM2$L_MAXFILES];
;  2753          4282  4          OUTPUT_ATTBUF[VSR_PROTECT] = .HOME_BLOCK[HM2$W_PROTECT];
;  2754          4283  4          OUTPUT_ATTBUF[VSR_FILEPROT] = .HOME_BLOCK[HM2$Q_FILEPROT];
;  2755          4284  4          OUTPUT_ATTBUF[VSR_RECPROT] = .HOME_BLOCK[HM2$W_RECPROT];
;  2756          4285  4          OUTPUT_ATTBUF[VSR_VOLCHAR] = .HOME_BLOCK[HM2$W_VOLCHAR];
;  2757          4286  4          OUTPUT_ATTBUF[VSR_EXTEND] = .HOME_BLOCK[HM2$W_EXTEND];
;  2758          4287  4          OUTPUT_ATTBUF[VSR_CLUSTER] = .HOME_BLOCK[HM2$Q_CLUSTER];
;  2759          4288  4          OUTPUT_ATTBUF[VSR_WINDOW] = .HOME_BLOCK[HM2$B_WINDOW];
;  2760          4289  4          OUTPUT_ATTBUF[VSR_LRU_LIM] = .HOME_BLOCK[HM2$B_LRU_LIM];
;  2761          4290  4          OUTPUT_ATTBUF[VSR_INDEXLBN] = .HOME_BLOCK[HM2$L_IBMAPLBN];
;  2762          4291  4          (OUTPUT_ATTBUF[VSR_RETAINMIN]) = .(HOME_BLOCK[HM2$Q_RETAINMIN]);
;  2763          4292  4          (OUTPUT_ATTBUF[VSR_RETAINMIN]+4) = .(HOME_BLOCK[HM2$Q_RETAINMIN]+4);
;  2764          4293  4          (OUTPUT_ATTBUF[VSR_RETAINMAX]) = .(HOME_BLOCK[HM2$Q_RETAINMAX]);
;  2765          4294  4          (OUTPUT_ATTBUF[VSR_RETAINMAX]+4) = .(HOME_BLOCK[HM2$Q_RETAINMAX]+4);
;  2766          4295  4          IF .HOME_BLOCK[HM2$W_RVN] EQL 1 AND .HOME_BLOCK[HM2$W_SETCOUNT] NEQ 0
;  2767          4296  4          THEN
;  2768          4297  4              CH$MOVE(
;  2769          4298  4                  HM2$S_STRUCNAME,
;  2770          4299  4                  HOME_BLOCK[HM2$T_STRUCNAME],
;  2771          4300  4                  COM_D_STRUCNAME);
;  2772          4301  4          END
;  2773          4302  3      ELSE
;  2774          4303  4          BEGIN
;  2775          4304  4          OUTPUT_ATTBUF[VSR_STRUCLEV] = .HOME_BLOCK[HM1$W_STRUCLEV];
;  2776          4305  4          BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$W_LENGTH] = HM1$S_VOLNAME;
;  2777          4306  4          BBLOCK[OUTPUT_ATTBUF[VSR_VOLNAME], DSC$A_POINTER] = HOME_BLOCK[HM1$T_VOLNAME2];
;  2778          4307  4          BBLOCK[OUTPUT_ATTBUF[VSR_OWNERNAME], DSC$W_LENGTH] = HM1$S_OWNERNAME;
;  2779          4308  4          BBLOCK[OUTPUT_ATTBUF[VSR_OWNERNAME], DSC$A_POINTER] = HOME_BLOCK[HM1$T_OWNERNAME];
;  2780          4309  4          FROM_ODS1_DATE(HOME_BLOCK[HM1$T_CREDATE], OUTPUT_ATTBUF[VSR_VOLDATE]);
;  2781          4310  4          OUTPUT_ATTBUF[VSR_VOLOWNER] = .(HOME_BLOCK[HM1$W_VOLOWNER])<0,8>;
;  2782          4311  4          (OUTPUT_ATTBUF[VSR_VOLOWNER])<16,8> = .(HOME_BLOCK[HM1$W_VOLOWNER])<8,8>;
;  2783          4312  4          OUTPUT_ATTBUF[VSR_MAXFILES] = .HOME_BLOCK[HM1$W_MAXFILES];
;  2784          4313  4          OUTPUT_ATTBUF[VSR_PROTECT] = .HOME_BLOCK[HM1$W_PROTECT];
;  2785          4314  4          OUTPUT_ATTBUF[VSR_FILEPROT] = .HOME_BLOCK[HM1$Q_FILEPROT];
;  2786          4315  4          OUTPUT_ATTBUF[VSR_EXTEND] = .HOME_BLOCK[HM1$B_EXTEND];
;  2787          4316  4          OUTPUT_ATTBUF[VSR_WINDOW] = .HOME_BLOCK[HM1$B_WINDOW];
;  2788          4317  4          OUTPUT_ATTBUF[VSR_LRU_LIM] = .HOME_BLOCK[HM1$B_LRU_LIM];
;  2789          4318  4          OUTPUT_ATTBUF[VSR_INDEXLBN] = ROT(.HOME_BLOCK[HM1$L_IBMAPLBN], 16);
;  2790          4319  3          END;
;  2791          4320  2      END;
;  2792          4321  2
;  2793          4322  2
;  2794          4323  2  ! Call the volume initialization routine to complete the work.
;  2795          4324  2  !
;  2796          4325  2  CURRENT_MTL[MTL_STRUCLEV] = .OUTPUT_ATTBUF[VSR_STRUCLEV];
;  2797          4326  2  CURRENT_VCB[VCB_ODS_2] = (.CURRENT_MTL[MTL_STRUCLEV] EQL 2);
;  2798          4327  2  INITIALIZE_VOLUME(.CURRENT_VCB, DEVCHAR);
;  2799          4328  1  END;


                                                        .EXTRN  SYS$GETCHN

                                    00FC 00000          .ENTRY  STA_INIVOL, Save R2,R3,R4,R5,R6,R7          ; 4168
                 57 00000000G   00  9E 00002          MOVAB   LIB$SIGNAL, R7                              ;
                 56 00000000'   EF  9E 00009          MOVAB   CURRENT_MTL, R6                             ;
                 5E     FD84    CE  9E 00010          MOVAB   -636(SP), SP                                ;
```

```
                                52      FB52    C6  3C 00015          MOVZWL  OUTPUT_ATTBUF+66, RVN                    ; 4208
                                                03  12 0001A          BNEQ    1$                                      ; 4209
                                52              01  D0 0001C          MOVL    #1, RVN                                 ;
                                66        FC    A6  D0 0001F 1$:     MOVL    OUTPUT_MTL, CURRENT_MTL                 ; 4210
                                1F      F996    C6  E8 00023         BLBS    QUAL+14, 4$                             ; 4216
                                50              66  D0 00028         MOVL    CURRENT_MTL, R0                         ; 4221
                                51      FA07    C6  9A 0002B         MOVZBL  COM_O_SETCOUNT, R1                      ; 4220
                                                06  12 00030         BNEQ    2$                                      ;
                                01        1F    A0  91 00032         CMPB    31(R0), #1                              ; 4221
                                                04  11 00036         BRB     3$                                      ;
                                51        1F    A0  91 00038 2$:     CMPB    31(R0), R1                              ; 4222
                                                09  13 0003C 3$:     BEQL    4$                                      ;
                            00000000G           8F  DD 0003E         PUSHL   #BACKUP$_BADSETCNT                      ; 4225
                                67              01  FB 00044         CALLS   #1, LIB$SIGNAL                          ;
                                50              66  D0 00047 4$:     MOVL    CURRENT_MTL, R0                         ; 4231
                                51        30    A0  9A 0004A         MOVZBL  48(R0), R1                              ;
                        51      52              51  C3 0004E         SUBL3   R1, RVN, R1                             ;
                51      1F  A0                  00  ED 00052         CMPZV   #0, #8, 31(R0), R1                      ;
                                                09  1E 00058         BGEQU   5$                                      ;
                            00000000G           8F  DD 0005A         PUSHL   #BACKUP$_INVATTVAL                      ; 4233
                                67              01  FB 00060         CALLS   #1, LIB$SIGNAL                          ;
                                51              66  D0 00063 5$:     MOVL    CURRENT_MTL, R1                         ; 4238
                                50        30    A1  9A 00066         MOVZBL  48(R1), R0                              ;
                        50      52              50  C3 0006A         SUBL3   R0, RVN, R0                             ;
                          04    A6        34 A140   D0 0006E         MOVL    52(R1)[R0], CURRENT_VCB                 ;
                                52              DD 00074             PUSHL   RVN                                     ; 4239
                      F222      CF              01  FB 00076         CALLS   #1, SWITCH_VOLUME                       ;
                                52              50  D0 0007B         MOVL    R0, CHANNEL                             ;
                          74    AE        74    8F  9A 0007E         MOVZBL  #116, DESC                              ; 4240
                          78    AE              6E  9E 00083         MOVAB   DEVCHAR, DESC+4                         ; 4241
                                                7E  7C 00087         CLRQ    -(SP)                                   ; 4242
                                          7C    AE  9F 00089         PUSHAB  DESC                                    ;
                                                7E  D4 0008C         CLRL    -(SP)                                   ;
                                                52  DD 0008E         PUSHL   CHANNEL                                 ;
                    00000000G   00              05  FB 00090         CALLS   #5, SYS$GETCHN                          ;
                                10              50  E8 00097         BLBS    STATUS, 6$                              ; 4243
                7E        04    A6              20  C1 0009A         ADDL3   #32, CURRENT_VCB, -(SP)                 ; 4245
                                                01  DD 0009F         PUSHL   #1                                      ;
                            00000000G           8F  DD 000A1         PUSHL   #BACKUP$_GETCHN                         ;
                                67              03  FB 000A7         CALLS   #3, LIB$SIGNAL                          ;
                03      F992    C6              05  E1 000AA 6$:     BBC     #5, QUAL+10, 7$                         ; 4251
                                          011B  31  000B0             BRW     11$                                     ;
                                          7C    AE  9F 000B3 7$:     PUSHAB  HOME_BLOCK                              ; 4257
                                          04    AE  9F 000B6         PUSHAB  DEVCHAR                                 ;
                                                52  DD 000B9         PUSHL   CHANNEL                                 ;
                        FDBA    CF              03  FB 000BB         CALLS   #3, READ_HOMEBLOCK                      ;
                        FB51    C6      0089    CE  91 000C0         CMPB    HOME_BLOCK+13, OUTPUT_ATTBUF+65         ; 4263
                                                10  13 000C7         BEQL    8$                                      ;
                7E        04    A6              20  C1 000C9         ADDL3   #32, CURRENT_VCB, -(SP)                 ; 4265
                                                01  DD 000CE         PUSHL   #1                                      ;
                            00000000G           8F  DD 000D0         PUSHL   #BACKUP$_STRUCLEV                       ;
                                67              03  FB 000D6         CALLS   #3, LIB$SIGNAL                          ;
                                02      0089    CE  91 000D9 8$:     CMPB    HOME_BLOCK+13, #2                       ; 4271
                                                03  13 000DE         BEQL    9$                                      ;
                                          0080  31  000E0             BRW     10$                                     ;
                        FB10    C6              0C  B0 000E3 9$:     MOVW    #12, OUTPUT_ATTBUF                      ; 4274
                        FB14    C6        D8    AD  9E 000E8         MOVAB   HOME_BLOCK+472, OUTPUT_ATTBUF+4         ; 4275
```

```
                FB18    C6              0C  B0 000EE           MOVW    #12, OUTPUT_ATTBUF+8                            ; 4276
                FB1C    C6        E4    AD  9E 000F3           MOVAB   HOME_BLOCK+484, OUTPUT_ATTBUF+12                ; 4277
                FB28    C6      00B8    CE  7D 000F9           MOVQ    HOME_BLOCK+60, OUTPUT_ATTBUF+24                 ; 4278
                FB38    C6      00A8    CE  D0 00100           MOVL    HOME_BLOCK+44, OUTPUT_ATTBUF+40                 ; 4280
                FB44    C6      0098    CE  D0 00107           MOVL    HOME_BLOCK+28, OUTPUT_ATTBUF+52                 ; 4281
                FB54    C6      00B0    CE  D0 0010E           MOVL    HOME_BLOCK+52, OUTPUT_ATTBUF+68                 ; 4282
                FB58    C6      00B4    CE  B0 00115           MOVW    HOME_BLOCK+56, OUTPUT_ATTBUF+72                 ; 4284
                FB5A    C6      00A6    CE  B0 0011C           MOVW    HOME_BLOCK+42, OUTPUT_ATTBUF+74                 ; 4285
                FB5C    C6      00C2    CE  B0 00123           MOVW    HOME_BLOCK+70, OUTPUT_ATTBUF+76                 ; 4286
                FB5E    C6      008A    CE  B0 0012A           MOVW    HOME_BLOCK+14, OUTPUT_ATTBUF+78                 ; 4287
                FB62    C6      00C0    CE  B0 00131           MOVW    HOME_BLOCK+68, OUTPUT_ATTBUF+82                 ; 4288
                FB64    C6      0094    CE  D0 00138           MOVL    HOME_BLOCK+24, OUTPUT_ATTBUF+84                 ; 4290
                FB70    C6      00C4    CE  7D 0013F           MOVQ    HOME_BLOCK+72, OUTPUT_ATTBUF+96                 ; 4291
                FB78    C6      00CC    CE  7D 00146           MOVQ    HOME_BLOCK+80, OUTPUT_ATTBUF+104                ; 4293
                        01      00A2    CE  B1 0014D           CMPW    HOME_BLOCK+38, #1                               ; 4295
                                        7A  12 00152           BNEQ    11$                                             ;
                                00A4    CE  B5 00154           TSTW    HOME_BLOCK+40                                   ;
                                        74  13 00158           BEQL    11$                                             ;
        FA14    C6        CC    AD          0C  28 0015A           MOVC3   #12, HOME_BLOCK+460, COM_O_STRUCNAME             ; 4299
                                            6B  11 00161           BRB     11$                                             ; 4271
                FB51    C6      0088    CE  90 00163 10$:      MOVB    HOME_BLOCK+12, OUTPUT_ATTBUF+65                 ; 4304
                FB10    C6              0C  B0 0016A           MOVW    #12, OUTPUT_ATTBUF                              ; 4305
                FB14    C6        D8    AD  9E 0016F           MOVAB   HOME_BLOCK+472, OUTPUT_ATTBUF+4                 ; 4306
                FB18    C6              0C  B0 00175           MOVW    #12, OUTPUT_ATTBUF+8                            ; 4307
                FB1C    C6        E4    AD  9E 0017A           MOVAB   HOME_BLOCK+484, OUTPUT_ATTBUF+12                ; 4308
                                FB28    C6  9F 00180           PUSHAB  OUTPUT_ATTBUF+24                                ; 4309
                                00BC    CE  9F 00184           PUSHAB  HOME_BLOCK+60                                   ;
            00000000G   00              02  FB 00188           CALLS   #2, FROM_ODS1_DATE                              ;
                FB38    C6      009A    CE  9A 0018F           MOVZBL  HOME_BLOCK+30, OUTPUT_ATTBUF+40                 ; 4310
                FB3A    C6      009B    CE  90 00196           MOVB    HOME_BLOCK+31, OUTPUT_ATTBUF+42                 ; 4311
                FB44    C6      0082    CE  3C 0019D           MOVZWL  HOME_BLOCK+6, OUTPUT_ATTBUF+52                  ; 4312
                FB54    C6      009C    CE  B0 001A4           MOVW    HOME_BLOCK+32, OUTPUT_ATTBUF+68                 ; 4313
                FB56    C6      00A0    CE  B0 001AB           MOVW    HOME_BLOCK+36, OUTPUT_ATTBUF+70                 ; 4314
                FB5C    C6      00A9    CE  9B 001B2           MOVZBW  HOME_BLOCK+45, OUTPUT_ATTBUF+76                 ; 4315
                FB62    C6      00A8    CE  90 001B9           MOVB    HOME_BLOCK+44, OUTPUT_ATTBUF+82                 ; 4316
                FB63    C6      00AA    CE  90 001C0           MOVB    HOME_BLOCK+46, OUTPUT_ATTBUF+83                 ; 4317
        FB64    C6        7E    AE          10  9C 001C7           ROTL    #16, HOME_BLOCK+2, OUTPUT_ATTBUF+84             ; 4318
                        50              66  D0 001CE 11$:       MOVL    CURRENT_MTL, R0                                 ; 4325
                      1E    A0  FB51    C6  90 001D1           MOVB    OUTPUT_ATTBUF+65, 30(R0)                        ;
                        52      04      A6  D0 001D7           MOVL    CURRENT_VCB, R2                                 ; 4326
                                        51  D4 001DB           CLRL    R1                                              ;
                        02      1E      A0  91 001DD           CMPB    30(R0), #2                                      ;
                                        02  12 001E1           BNEQ    12$                                             ;
                                        51  D6 001E3           INCL    R1                                              ;
    07  A2          01          01      51  F0 001E5 12$:       INSV    R1, #1, #1, 7(R2)                               ;
                                4004    8F  BB 001EB           PUSHR   #^M<R2,SP>                                      ; 4327
                00000000G   00          02  FB 001EF           CALLS   #2, INITIALIZE_VOLUME                           ;
                                        04  001F6           RET                                                     ; 4328

; Routine Size:  503 bytes,    Routine Base:  CODE + 0EA4
```

```
;  2801          4329  1 %SBTTL 'STA_INIT_HDRS - initialize volume file headers'
;  2802          4330  1 GLOBAL ROUTINE STA_INIT_HDRS (REC): NOVALUE=
;  2803          4331  1
;  2804          4332  1 !++
;  2805          4333  1 !
;  2806          4334  1 ! FUNCTIONAL DESCRIPTION:
;  2807          4335  1 !       This routine initializes file headers on an output volume to have the
;  2808          4336  1 !       file sequence numbers that they had originally.
;  2809          4337  1 !
;  2810          4338  1 ! INPUT PARAMETERS:
;  2811          4339  1 !       REC             - Pointer to save set record, type BRH$K_FID.
;  2812          4340  1 !
;  2813          4341  1 ! IMPLICIT INPUTS:
;  2814          4342  1 !       OUTPUT_MTL      - Pointer to MTL for output volume set.
;  2815          4343  1 !
;  2816          4344  1 ! OUTPUT PARAMETERS:
;  2817          4345  1 !       NONE
;  2818          4346  1 !
;  2819          4347  1 ! IMPLICIT OUTPUTS:
;  2820          4348  1 !       NONE
;  2821          4349  1 !
;  2822          4350  1 ! ROUTINE VALUE:
;  2823          4351  1 !       NONE
;  2824          4352  1 !
;  2825          4353  1 ! SIDE EFFECTS:
;  2826          4354  1 !       NONE
;  2827          4355  1 !
;  2828          4356  1 !--
;  2829          4357  1
;  2830          4358  2 BEGIN
;  2831          4359  2 MAP
;  2832          4360  2         REC:            REF BBLOCK;             ! Pointer to save set record
;  2833          4361  2 LOCAL
;  2834          4362  2         RSA:            VECTOR[NAM$C_MAXRSS,BYTE],      ! Resultant string area
;  2835          4363  2         CHANNEL,                                ! Channel for output volume
;  2836          4364  2         STATUS,                                 ! Status return
;  2837          4365  2         IOSB:           VECTOR[4,WORD],         ! I/O status block
;  2838          4366  2         FILE_ID:        BBLOCK[FID$C_LENGTH],   ! Current file ID
;  2839          4367  2         FILE_NUMBER,                            ! File number of first header
;  2840          4368  2         AREA,                                   ! Pointer to allocated memory
;  2841          4369  2         HDR:            REF BBLOCK,             ! Pointer to current header
;  2842          4370  2         SEQ:            REF BBLOCK;             ! Cursor for sequence vector
;  2843          4371  2
;  2844          4372  2
;  2845          4373  2 ! Initialize.
;  2846          4374  2 !
;  2847          4375  2 CURRENT_MTL = .OUTPUT_MTL;
;  2848          4376  2 IF .REC[BSA$B_FID_RVN] GEQU .CURRENT_MTL[MTL_SETCOUNT] + .CURRENT_MTL[MTL_RVN_BASE]
;  2849          4377  2 OR .REC[BSA$B_FID_RVN] LSSU .CURRENT_MTL[MTL_RVN_BASE]
;  2850          4378  2 THEN SIGNAL(BACKUP$_INVATTVAL);
;  2851          4379  2 CURRENT_VCB = .CURRENT_MTL[MTL_VCB(.REC[BSA$B_FID_RVN]-.CURRENT_MTL[MTL_RVN_BASE])];
;  2852          4380  2 RSA_DESC[0] = NAM$C_MAXRSS;
;  2853          4381  2 RSA_DESC[1] = RSA;
;  2854        P 4382  2 $FAO(
;  2855        P 4383  2     $DESCRIPTOR('!AS[000000]INDEXF.SYS;1'),
;  2856        P 4384  2     RSA_DESC,
;  2857        P 4385  2     RSA_DESC,
```

```
;  2858          4386  2     CURRENT_VCB[VCB_DEVICE]);
;  2859          4387  2
;  2860          4388  2
;  2861          4389  2 ! Get memory for buffer.
;  2862          4390  2 !
;  2863          4391  2 AREA = GET_VM(512 * .REC[BSA$W_FID_COUNT]);
;  2864          4392  2
;  2865          4393  2
;  2866          4394  2 HDR = .AREA;
;  2867          4395  2 SEQ = .REC;
;  2868          4396  2 FILE_NUMBER = .REC[BSA$W_FID_NUM];
;  2869          4397  2 FILE_NUMBER<16,8> = .REC[BSA$B_FID_NMX];
;  2870          4398  2 INCR N FROM .FILE_NUMBER TO .FILE_NUMBER+.REC[BSA$W_FID_COUNT]-1 DO
;  2871          4399  3     BEGIN
;  2872          4400  3     IF .N GTRU .CURRENT_VCB[VCB_MAXFILIDX] THEN SIGNAL(BACKUP$_INVATTVAL);
;  2873          4401  3     FILE_ID[FID$W_NUM] = .N;
;  2874          4402  3     FILE_ID[FID$B_NMX] = .N<16,8>;
;  2875          4403  3     FILE_ID[FID$W_SEQ] = .SEQ[BSA$W_FID_SEQ];
;  2876          4404  3     FILE_ID[FID$B_RVN] = .REC[BSA$B_FID_RVN];
;  2877          4405  3     IF
;  2878          4406  4         BEGIN
;  2879          4407  4         IF .N-1 LEQU 15
;  2880          4408  4         THEN
;  2881          4409  4             .BITVECTOR[CURRENT_VCB[VCB_INIT_HDRS], .N-1]
;  2882          4410  4         ELSE
;  2883          4411  4             FALSE
;  2884          4412  4         END
;  2885          4413  3     THEN
;  2886          4414  4         BEGIN
;  2887          4415  4
;  2888          4416  4         ! Initial file header.  Reread it, rather than blowing it away.
;  2889          4417  4         !
;  2890          4418  4         READ_HEADER(FILE_ID, .HDR);
;  2891          4419  4         END
;  2892          4420  3     ELSE
;  2893          4421  4         BEGIN
;  2894          4422  4
;  2895          4423  4         ! Initialize file header as a deleted header with the appropriate
;  2896          4424  4         ! file sequence number.
;  2897          4425  4         !
;  2898          4426  4         CREATE_DELHDR(FILE_ID, .HDR);
;  2899          4427  3         END;
;  2900          4428  3     HDR = .HDR + 512;
;  2901          4429  3     SEQ = .SEQ + 2;
;  2902          4430  2     END;
;  2903          4431  2
;  2904          4432  2
;  2905          4433  2 ! Write out buffer to index file.
;  2906          4434  2 !
;  2907          4435  2 CHANNEL = SWITCH_VOLUME(.REC[BSA$B_FID_RVN]);
;  2908          4436  2 CURRENT_WCB = .CURRENT_VCB[VCB_INDEXF];
;  2909          4437  2 STATUS = R_W_VIRTUAL(
;  2910          4438  2     0,
;  2911          4439  2     0,
;  2912          4440  2     IO$_WRITEVBLK,
;  2913          4441  2     IOSB,
;  2914          4442  2     0,
```

```
;  2915          4443  2     0,
;  2916          4444  2     .AREA,
;  2917          4445  2     512 * .REC[BSA$W_FID_COUNT],
;  2918          4446  2     .FILE_NUMBER + .CURRENT_VCB[VCB_HDR_OFFSET]);
;  2919          4447  2 $WAITFR(EFN=0);
;  2920          4448  2 IF .STATUS THEN STATUS = .IOSB[0];
;  2921          4449  2 IF NOT .STATUS
;  2922          4450  2 THEN
;  2923          4451  2     SIGNAL(BACKUP$_WRITEERR + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;  2924          4452  2
;  2925          4453  2
;  2926          4454  2 ! Release memory for buffer.
;  2927          4455  2 !
;  2928          4456  2 FREE_VM(512 * .REC[BSA$W_FID_COUNT], .AREA);
;  2929          4457  1 END;


45  44  4E  49  5D  30  30  30  30  30  30  5B  53  41  21  0109B P.AAD:  .ASCII  \!AS[000000]INDEXF.SYS;1\                    ;
                            31  3B  53  59  53  2E  46  58  010AA                                                          ;
                                                            010B2        .BLKB   2                                         ;
                                                00000017    010B4 P.AAC: .LONG   23                                        ;
                                                00000000'   010B8        .ADDRESS P.AAD                                    ;

                                                                         .EXTRN  SYS$FAO

                                                    0FFC 00000           .ENTRY  STA_INIT_HDRS, Save R2,R3,R4,R5,R6,R7,R8,-  ; 4330
                                                                                 R9,R10,R11                                ;
                                5B 00000000'    EF  9E 00002             MOVAB   RSA_DESC, R11                             ;
                                5A 00000000'    EF  9E 00009             MOVAB   CURRENT_VCB, R10                          ;
                                5E      FEF0    CE  9E 00010             MOVAB   -272(SP), SP                              ;
                            FC  AA      F8  AA  D0 00015                 MOVL    OUTPUT_MTL, CURRENT_MTL                   ; 4375
                                53      04  AC  D0 0001A                 MOVL    REC, R3                                   ; 4376
                                56      04  A3  9A 0001E                 MOVZBL  4(R3), R6                                 ;
                                50      FC  AA  D0 00022                 MOVL    CURRENT_MTL, R0                           ;
                                51      1F  A0  9A 00026                 MOVZBL  31(R0), R1                                ;
                                52      30  A0  9A 0002A                 MOVZBL  48(R0), R2                                ;
                                51          52  C0 0002E                 ADDL2   R2, R1                                    ;
                                51          56  D1 00031                 CMPL    R6, R1                                    ;
                                            06  1E 00034                 BGEQU   1$                                        ;
                                56      30  A0  91 00036                 CMPB    48(R0), R6                                ; 4377
                                            0D  1B 0003A                 BLEQU   2$                                        ;
                                00000000G   8F  DD 0003C 1$:             PUSHL   #BACKUP$_INVATTVAL                        ; 4378
                        00000000G   00      01  FB 00042                 CALLS   #1, LIB$SIGNAL                            ;
                                51      FC  AA  D0 00049 2$:             MOVL    CURRENT_MTL, R1                           ; 4379
                                50      30  A1  9A 0004D                 MOVZBL  48(R1), R0                                ;
                    50          56          50  C3 00051                 SUBL3   R0, R6, R0                                ;
                                6A      34 A140 D0 00055                 MOVL    52(R1)[R0], CURRENT_VCB                   ;
                                6B      FF  8F  9A 0005A                 MOVZBL  #255, RSA_DESC                            ; 4380
                            04  AB      10  AE  9E 0005E                 MOVAB   RSA, RSA_DESC+4                           ; 4381
                    7E          6A          20  C1 00063                 ADDL3   #32, CURRENT_VCB, -(SP)                   ; 4386
                                            5B  DD 00067                 PUSHL   R11                                       ;
                                            5B  DD 00069                 PUSHL   R11                                       ;
                                        8A  AF  9F 0006B                 PUSHAB  P.AAC                                     ;
                        00000000G   00          04  FB 0006E             CALLS   #4, SYS$FAO                               ;
                                55      06  A3  3C 00075                 MOVZWL  6(R3), R5                                 ; 4391
                    57          55          09  78 00079                 ASHL    #9, R5, R7                                ;
```

```
                                                 57  DD 0007D          PUSHL   R7
                                00000000G   00   01  FB 0007F          CALLS   #1, GET_VM
                                            59   50  D0 00086          MOVL    R0, AREA
                                            58   59  D0 00089          MOVL    AREA, HDR                       4394
                                            54   53  D0 0008C          MOVL    R3, SEQ                         4395
                                            52   02  A3 3C 0008F       MOVZWL  2(R3), FILE_NUMBER              4396
                52              08          10   05  A3 F0 00093       INSV    5(R3), #16, #8, FILE_NUMBER     4397
                53              52          52   55  C1 00099          ADDL3   R5, FILE_NUMBER, R3             4398
                                            55   FF  A2 9E 0009D       MOVAB   -1(R2), N
                                                     5A  11 000A1       BRB     7$
                                            50       6A  D0 000A3  3$:  MOVL    CURRENT_VCB, R0                 4400
                                        1C  A0       55  D1 000A6       CMPL    N, 28(R0)
                                                     0D  1B 000AA       BLEQU   4$
                                    00000000G    8F  DD 000AC       PUSHL   #BACKUP$_INVATTVAL
                                00000000G   00       01  FB 000B2       CALLS   #1, LIB$SIGNAL
                                            6E       55  B0 000B9  4$:  MOVW    N, FILE_ID                      4401
                50              55          08       10  EF 000BC       EXTZV   #16, #8, N, R0                  4402
                                        05  AE       50  90 000C1       MOVB    R0, FILE_ID+5
                                        02  AE   08  A4  B0 000C5       MOVW    8(SEQ), FILE_ID+2               4403
                                        04  AE       56  90 000CA       MOVB    R6, FILE_ID+4                   4404
                                            51   FF  A5  9E 000CE       MOVAB   -1(R5), R1                      4407
                                            0F       51  D1 000D2       CMPL    R1, #15
                                                     14  1A 000D5       BGTRU   5$
                                            50       6A  D0 000D7       MOVL    CURRENT_VCB, R0                 4409
                            0C          18  A0       51  E1 000DA       BBC     R1, 24(R0), 5$
                                                     58  DD 000DF       PUSHL   HDR                             4418
                                            04   AE  9F 000E1       PUSHAB  FILE_ID
                                    F10B    CF       02  FB 000E4       CALLS   #2, READ_HEADER
                                                     0A  11 000E9       BRB     6$                              4405
                                                     58  DD 000EB  5$:  PUSHL   HDR                             4426
                                            04   AE  9F 000ED       PUSHAB  FILE_ID
                                    F281    CF       02  FB 000F0       CALLS   #2, CREATE_DELHDR
                                            58   0200    C8  9E 000F5  6$:  MOVAB   512(R8), HDR                 4428
                                            54       02  C0 000FA       ADDL2   #2, SEQ                         4429
                        A2                  55       53  F2 000FD  7$:  AOBLSS  R3, N, 3$                       4398
                                                     56  DD 00101       PUSHL   R6                              4435
                                    EF7D    CF       01  FB 00103       CALLS   #1, SWITCH_VOLUME
                                            50       6A  D0 00108       MOVL    CURRENT_VCB, R0                 4436
                                        04  AA       60  D0 0010B       MOVL    (R0), CURRENT_WCB
                                            50   1A      A0  3C 0010F   MOVZWL  26(R0), R0                     4446
                                                     6042    9F 00113   PUSHAB  (R0)[FILE_NUMBER]
                                                     57  DD 00116       PUSHL   R7                              4445
                                                     59  DD 00118       PUSHL   AREA                            4444
                                                     7E  7C 0011A       CLRQ    -(SP)                           4437
                                                 1C  AE  9F 0011C       PUSHAB  IOSB
                                                     30  DD 0011F       PUSHL   #48
                                                     7E  7C 00121       CLRQ    -(SP)
                                    F908    CF       09  FB 00123       CALLS   #9, R_W_VIRTUAL
                                            52       50  D0 00128       MOVL    R0, STATUS
                                                     7E  D4 0012B       CLRL    -(SP)                           4447
                                00000000G   00       01  FB 0012D       CALLS   #1, SYS$WAITFR
                                            07       52  E9 00134       BLBC    STATUS, 8$                      4448
                                            52   08  AE  3C 00137       MOVZWL  IOSB, STATUS
                                            13       52  E8 0013B       BLBS    STATUS, 9$                      4449
                                                     52  DD 0013E  8$:  PUSHL   STATUS                          4451
                                                     5B  DD 00140       PUSHL   R11
                                                     01  DD 00142       PUSHL   #1
```

```
                                 00000000G  8F  DD 00144          PUSHL   #BACKUP$_WRITEERR+2                    ;
                    00000000G  00            04  FB 0014A          CALLS   #4, LIB$SIGNAL                        ;
                                      0280    8F  BB 00151 9$:     PUSHR   #^M<R7,R9>                            ; 4456
                    00000000G  00            02  FB 00155          CALLS   #2, FREE_VM                           ;
                                             04  0015C          RET                                           ; 4457

; Routine Size:  349 bytes,    Routine Base:  CODE + 10BC
```

```
;2931          4458  1 %SBTTL 'STA_WRITEBOOT - write volume boot block'
;2932          4459  1 GLOBAL ROUTINE STA_WRITEBOOT : NOVALUE=
;2933          4460  1
;2934          4461  1 !++
;2935          4462  1 !
;2936          4463  1 ! FUNCTIONAL DESCRIPTION:
;2937          4464  1 !       This routine is called to initialize the boot block.
;2938          4465  1 !
;2939          4466  1 ! INPUT PARAMETERS:
;2940          4467  1 !       NONE
;2941          4468  1 !
;2942          4469  1 ! IMPLICIT INPUTS:
;2943          4470  1 !       OUTPUT_ATTBUF   - Contains file attributes.
;2944          4471  1 !       OUTPUT_MTL      - Pointer to MTL for output volume set.
;2945          4472  1 !                         Boot file is accessed.
;2946          4473  1 !
;2947          4474  1 ! OUTPUT PARAMETERS:
;2948          4475  1 !       NONE
;2949          4476  1 !
;2950          4477  1 ! IMPLICIT OUTPUTS:
;2951          4478  1 !       NONE
;2952          4479  1 !
;2953          4480  1 ! ROUTINE VALUE:
;2954          4481  1 !       NONE
;2955          4482  1 !
;2956          4483  1 ! SIDE EFFECTS:
;2957          4484  1 !       Boot block rewritten.
;2958          4485  1 !
;2959          4486  1 !--
;2960          4487  1
;2961          4488  2 BEGIN
;2962          4489  2 LOCAL
;2963          4490  2         RSA:            VECTOR[NAM$C_MAXRSS,BYTE],      ! Resultant string area
;2964          4491  2         BUFFER:         BBLOCK[512],    ! Buffer for boot block
;2965          4492  2         STATUS,                         ! Status variable
;2966          4493  2         IOSB:           VECTOR[4,WORD]; ! I/O status block
;2967          4494  2
;2968          4495  2
;2969          4496  2 ! Read the boot block from the relative volume on which the presently accessed
;2970          4497  2 ! file is located.
;2971          4498  2 !
;2972          4499  2 CURRENT_MTL = .OUTPUT_MTL;
;2973          4500  2 CURRENT_VCB = .CURRENT_MTL[MTL_VCB(.CURRENT_MTL[MTL_FID_RVN]-.CURRENT_MTL[MTL_RVN_BASE])];
;2974          4501  2 CURRENT_WCB = .CURRENT_VCB[VCB_INDEXF];
;2975          4502  2 RSA_DESC[0] = NAM$C_MAXRSS;
;2976          4503  2 RSA_DESC[1] = RSA;
;2977        P 4504  2 $FAO(
;2978        P 4505  2     $DESCRIPTOR('!AS[000000]INDEXF.SYS;1'),
;2979        P 4506  2     RSA_DESC,
;2980        P 4507  2     RSA_DESC,
;2981          4508  2     CURRENT_VCB[VCB_DEVICE]);
;2982          4509  2 STATUS = R_W_VIRTUAL(
;2983          4510  2     0,
;2984          4511  2     0,
;2985          4512  2     IO$_READVBLK,
;2986          4513  2     IOSB,
;2987          4514  2     0,
```

```
;  2988          4515  2      0,
;  2989          4516  2      BUFFER,
;  2990          4517  2      512,
;  2991          4518  2      1);
;  2992          4519  2  $WAITFR(EFN=0);
;  2993          4520  2  IF .STATUS THEN STATUS = .IOSB[0];
;  2994          4521  2  IF NOT .STATUS
;  2995          4522  2  THEN
;  2996          4523  3      BEGIN
;  2997          4524  3      SIGNAL(BACKUP$_READERR + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;  2998          4525  3      RETURN;
;  2999          4526  2      END;
;  3000          4527  2
;  3001          4528  2
;  3002          4529  2  ! Update the boot LBN in the second longword of the boot block to point to
;  3003          4530  2  ! the VBN given by OUTPUT_ATTBUF[FAR_BOOTVBN] of the presently accessed file.
;  3004          4531  2  ! The file is assumed to be contiguous.
;  3005          4532  2  !
;  3006          4533  2  BUFFER[4,0,32,0] =
;  3007          4534  2      ROT(
;  3008          4535  2          .BBLOCK[BBLOCK[.CURRENT_MTL[MTL_WINDOW], WCB_S_HEADER,0,0,0], WCB_LBN] +
;  3009          4536  2          .OUTPUT_ATTBUF[FAR_BOOTVBN] - 1,
;  3010          4537  2      16);
;  3011          4538  2
;  3012          4539  2
;  3013          4540  2  ! Rewrite the boot block.
;  3014          4541  2  !
;  3015          4542  2  STATUS = R_W_VIRTUAL(
;  3016          4543  2      0,
;  3017          4544  2      0,
;  3018          4545  2      IO$_WRITEVBLK,
;  3019          4546  2      IOSB,
;  3020          4547  2      0,
;  3021          4548  2      0,
;  3022          4549  2      BUFFER,
;  3023          4550  2      512,
;  3024          4551  2      1);
;  3025          4552  2  $WAITFR(EFN=0);
;  3026          4553  2  IF .STATUS THEN STATUS = .IOSB[0];
;  3027          4554  2  IF NOT .STATUS
;  3028          4555  2  THEN
;  3029          4556  2      SIGNAL(BACKUP$_WRITEERR + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;  3030          4557  1  END;


45  44  4E  49  5D  30  30  30  30  30  30  5B  53  41  21  01219 P.AAF:  .ASCII  \!AS[000000]INDEXF.SYS;1\                    ;
                            31  3B  53  59  53  2E  46  58  01228
                                                00000017  01230 P.AAE:  .LONG   23                                         ;
                                                00000000' 01234        .ADDRESS P.AAF                                     ;


                                                    007C 00000         .ENTRY  STA_WRITEBOOT, Save R2,R3,R4,R5,R6          ; 4459
                                    56      F8AE    CF  9E 00002       MOVAB   R_W_VIRTUAL, R6                             ;
                                    55 00000000G    00  9E 00007       MOVAB   SYS$WAITFR, R5                              ;
                                    54 00000000'    EF  9E 0000E       MOVAB   RSA_DESC, R4                                ;
```

```
                                53 00000000'   EF   9E 00015          MOVAB   CURRENT_MTL, R3                                    ;
                                5E     FCF8    CE   9E 0001C          MOVAB   -776(SP), SP                                      ;
                                63       FC    A3   D0 00021          MOVL    OUTPUT_MTL, CURRENT_MTL                           ; 4499
                                51             63   D0 00025          MOVL    CURRENT_MTL, R1                                   ; 4500
                                50       1C    A1   9A 00028          MOVZBL  28(R1), R0                                        ;
                                52       30    A1   9A 0002C          MOVZBL  48(R1), R2                                        ;
                                50             52   C2 00030          SUBL2   R2, R0                                            ;
                          04    A3       34 A140   D0 00033          MOVL    52(R1)[R0], CURRENT_VCB                           ;
                          08    A3       04   B3   D0 00039          MOVL    @CURRENT_VCB, CURRENT_WCB                         ; 4501
                                64       FF   8F   9A 0003E          MOVZBL  #255, RSA_DESC                                    ; 4502
                          04    A4     FF00   CD   9E 00042          MOVAB   RSA, RSA_DESC+4                                   ; 4503
                    7E    04    A3             20   C1 00048          ADDL3   #32, CURRENT_VCB, -(SP)                           ; 4508
                                               54   DD 0004D          PUSHL   R4                                                ;
                                               54   DD 0004F          PUSHL   R4                                                ;
                                         A4    AF   9F 00051          PUSHAB  P.AAE                                             ;
                          00000000G     00         04   FB 00054          CALLS   #4, SYS$FAO                                       ;
                                               01   DD 0005B          PUSHL   #1                                                ; 4509
                                7E     0200    8F   3C 0005D          MOVZWL  #512, -(SP)                                       ;
                                         10    AE   9F 00062          PUSHAB  BUFFER                                            ;
                                               7E   7C 00065          CLRQ    -(SP)                                             ;
                                         14    AE   9F 00067          PUSHAB  IOSB                                              ;
                                               31   DD 0006A          PUSHL   #49                                               ;
                                               7E   7C 0006C          CLRQ    -(SP)                                             ;
                                66             09   FB 0006E          CALLS   #9, R_W_VIRTUAL                                   ;
                                52             50   D0 00071          MOVL    R0, STATUS                                        ;
                                               7E   D4 00074          CLRL    -(SP)                                             ; 4519
                                65             01   FB 00076          CALLS   #1, SYS$WAITFR                                    ;
                                06             52   E9 00079          BLBC    STATUS, 1$                                        ; 4520
                                52             6E   3C 0007C          MOVZWL  IOSB, STATUS                                      ;
                                0E             52   E8 0007F          BLBS    STATUS, 2$                                        ; 4521
                                               52   DD 00082 1$:       PUSHL   STATUS                                            ; 4524
                                               54   DD 00084          PUSHL   R4                                                ;
                                               01   DD 00086          PUSHL   #1                                                ;
                                00000000G      8F   DD 00088          PUSHL   #BACKUP$_READERR+2                                ;
                                               48   11 0008E          BRB     4$                                                ;
                                50             63   D0 00090 2$:       MOVL    CURRENT_MTL, R0                                   ; 4535
                                50       08    A0   D0 00093          MOVL    8(R0), R0                                         ;
                    50          18    A0     FB78   C3   C1 00097          ADDL3   OUTPUT_ATTBUF+104, 24(R0), R0                     ; 4536
                                               50   D7 0009E          DECL    R0                                                ;
                OC  AE                         50   10   9C 000A0          ROTL    #16, R0, BUFFER+4                                 ; 4534
                                               01   DD 000A5          PUSHL   #1                                                ; 4542
                                7E     0200    8F   3C 000A7          MOVZWL  #512, -(SP)                                       ;
                                         10    AE   9F 000AC          PUSHAB  BUFFER                                            ;
                                               7E   7C 000AF          CLRQ    -(SP)                                             ;
                                         14    AE   9F 000B1          PUSHAB  IOSB                                              ;
                                               30   DD 000B4          PUSHL   #48                                               ;
                                               7E   7C 000B6          CLRQ    -(SP)                                             ;
                                66             09   FB 000B8          CALLS   #9, R_W_VIRTUAL                                   ;
                                52             50   D0 000BB          MOVL    R0, STATUS                                        ;
                                               7E   D4 000BE          CLRL    -(SP)                                             ; 4552
                                65             01   FB 000C0          CALLS   #1, SYS$WAITFR                                    ;
                                06             52   E9 000C3          BLBC    STATUS, 3$                                        ; 4553
                                52             6E   3C 000C6          MOVZWL  IOSB, STATUS                                      ;
                                13             52   E8 000C9          BLBS    STATUS, 5$                                        ; 4554
                                               52   DD 000CC 3$:       PUSHL   STATUS                                            ; 4556
                                               54   DD 000CE          PUSHL   R4                                                ;
                                               01   DD 000D0          PUSHL   #1                                                ;
```

```
                                00000000G  8F  DD 000D2          PUSHL   #BACKUP$_WRITEERR+2                      ;
                     00000000G  00          04  FB 000D8 4$:     CALLS   #4, LIB$SIGNAL                          ;
                                            04 000DF 5$:      RET                                               ; 4557

; Routine Size:  224 bytes,    Routine Base:  CODE + 1238
```

```
;3032        4558  1 %SBTTL 'STA_MOUNT - mount volume for stand-alone use'
;3033        4559  1 GLOBAL ROUTINE STA_MOUNT (MODE, P_RVN): NOVALUE=
;3034        4560  1
;3035        4561  1 !++
;3036        4562  1 !
;3037        4563  1 ! FUNCTIONAL DESCRIPTION:
;3038        4564  1 !       This routine mounts a volume set.
;3039        4565  1 !
;3040        4566  1 ! INPUT PARAMETERS:
;3041        4567  1 !       MODE            - 0: input volume to be read
;3042        4568  1 !                         1: output volume to be initialized
;3043        4569  1 !                         3: output volume to be updated
;3044        4570  1 !       P_RVN (optional): if absent, completely mount entire volume set
;3045        4571  1 !                         if n, mount RVN n of sequential save set
;3046        4572  1 !
;3047        4573  1 ! IMPLICIT INPUTS:
;3048        4574  1 !       NONE
;3049        4575  1 !
;3050        4576  1 ! OUTPUT PARAMETERS:
;3051        4577  1 !       NONE
;3052        4578  1 !
;3053        4579  1 ! IMPLICIT OUTPUTS:
;3054        4580  1 !       INPUT_MTL       - Pointer to MTL for input volume set.
;3055        4581  1 !       OUTPUT_MTL      - Pointer to MTL for output volume set.
;3056        4582  1 !
;3057        4583  1 ! ROUTINE VALUE:
;3058        4584  1 !       NONE
;3059        4585  1 !
;3060        4586  1 ! SIDE EFFECTS:
;3061        4587  1 !       NONE
;3062        4588  1 !
;3063        4589  1 !--
;3064        4590  1
;3065        4591  2 BEGIN
;3066        4592  2
;3067        4593  2 LINKAGE
;3068        4594  2         L_MAP_POINTER=  JSB:
;3069        4595  2                         GLOBAL(COUNT=6, LBN=7, MAP_POINTER=8);
;3070        4596  2 LOCAL
;3071        4597  2         Q:              REF BBLOCK,     ! Pointer to qualifier block
;3072        4598  2         MTL:            REF BBLOCK,     ! Pointer to MTL block
;3073        4599  2         VCB:            REF BBLOCK,     ! Pointer to VCB block
;3074        4600  2         SETCOUNT,                       ! Count of volumes in set
;3075        4601  2         CHANNEL,                        ! Channel number
;3076        4602  2         STATUS,                         ! System service status
;3077        4603  2         IOSB:           VECTOR[4,WORD], ! I/O status block
;3078        4604  2         DEVICE_CHAR:    BBLOCK[DIB$C_LENGTH],   ! Device characteristics
;3079        4605  2         DESC:           VECTOR[2],      ! Descriptor
;3080        4606  2         HOME_BLOCK:     BBLOCK[512],    ! Home block buffer
;3081        4607  2         HEADER:         BBLOCK[512];    ! Index file header buffer
;3082        4608  2 BIND
;3083        4609  2         INDEX_FILE_ID = UPLIT WORD (FID$C_INDEXF, FID$C_INDEXF, 0),
;3084        4610  2         BITMAP_FILE_ID = UPLIT WORD (FID$C_BITMAP, FID$C_BITMAP, 0);
;3085        4611  2 SWITCHES
;3086        4612  2         NOSAFE;
;3087        4613  2 BUILTIN
;3088        4614  2         ACTUALCOUNT;
```

```
;  3089          4615  2 EXTERNAL ROUTINE
;  3090          4616  2         GET_MAP_POINTER: L_MAP_POINTER; ! Get value of ODS-2 file map pointer
;  3091          4617  2
;  3092          4618  2
;  3093          4619  2 ! Count the list of devices.
;  3094          4620  2 !
;  3095          4621  3 IF (IF .MODE THEN .OUTPUT_MTL EQL 0 ELSE .INPUT_MTL EQL 0)
;  3096          4622  2 THEN
;  3097          4623  3     BEGIN
;  3098          4624  3     Q = (IF .MODE THEN .QUAL[QUAL_OUTP_LIST] ELSE .QUAL[QUAL_INPU_LIST]);
;  3099          4625  3     SETCOUNT = 0;
;  3100          4626  3     WHILE .Q NEQ 0 DO
;  3101          4627  4         BEGIN
;  3102          4628  4         SETCOUNT = .SETCOUNT + 1;
;  3103          4629  4         Q = .Q[QUAL_NEXT];
;  3104          4630  3         END;
;  3105          4631  3
;  3106          4632  3
;  3107          4633  3 ! Allocate the MTL and the VCB's.
;  3108          4634  3 !
;  3109          4635  3     MTL = GET_ZERO_VM(MTL_S_ENTRY + .SETCOUNT * %UPVAL);
;  3110          4636  3     CURRENT_MTL = .MTL;
;  3111          4637  3     IF .MODE
;  3112          4638  3     THEN
;  3113          4639  4         BEGIN
;  3114          4640  4         OUTPUT_MTL = .MTL;
;  3115          4641  4         IF ACTUALCOUNT () GEQ 2 THEN MTL[MTL_SEQ_DISK] = 1;
;  3116          4642  4         END
;  3117          4643  3     ELSE
;  3118          4644  3         INPUT_MTL = .MTL;
;  3119          4645  3     MTL[MTL_SETCOUNT] = .SETCOUNT;
;  3120          4646  3     MTL[MTL_RVN_BASE] = 1;
;  3121          4647  3     MTL[MTL_HEADER] = GET_VM(512);
;  3122          4648  3
;  3123          4649  3 ! Set up the skeleton VCB's
;  3124          4650  3 !
;  3125          4651  3
;  3126          4652  3     Q = (IF .MODE THEN .QUAL[QUAL_OUTP_LIST] ELSE .QUAL[QUAL_INPU_LIST]);
;  3127          4653  3     INCR RVN FROM 1 TO .SETCOUNT DO
;  3128          4654  4         BEGIN
;  3129          4655  4         MTL[MTL_VCB(.RVN-1)] = VCB = GET_ZERO_VM(VCB_S_ENTRY);
;  3130          4656  4         VCB[VCB_RVN] = .RVN;
;  3131          4657  4         VCB[VCB_ACB_FLINK] = VCB[VCB_ACB_BLINK] = VCB[VCB_ACB_FLINK];
;  3132          4658  4         IF .MODE NEQ 0 THEN VCB[VCB_OUTPUT] = TRUE;
;  3133          4659  4         VCB[VCB_FAB] = .Q[QUAL_PARA_FC];
;  3134          4660  4         BBLOCK[VCB[VCB_DEVICE], DSC$W_LENGTH] =
;  3135          4661  4             .BBLOCK[Q[QUAL_DEV_DESC], DSC$W_LENGTH];
;  3136          4662  4         BBLOCK[VCB[VCB_DEVICE], DSC$A_POINTER] =
;  3137          4663  4             .BBLOCK[Q[QUAL_DEV_DESC], DSC$A_POINTER];
;  3138          4664  4         Q = .Q[QUAL_NEXT];
;  3139          4665  3         END;
;  3140          4666  2     END;
;  3141          4667  2
;  3142          4668  2 ! Now go back and do the actual mount for those volumes requested.
;  3143          4669  2 !
;  3144          4670  2
;  3145          4671  2 IF .MODE
```

```
; 3146      4672  2 THEN CURRENT_MTL = MTL = .OUTPUT_MTL
; 3147      4673  2 ELSE CURRENT_MTL = MTL = .INPUT_MTL;
; 3148      4674  2 SETCOUNT = .MTL[MTL_SETCOUNT];
; 3149      4675  2 IF ACTUALCOUNT () LSS 2
; 3150      4676  2 OR .P_RVN NEQ 0
; 3151      4677  2 THEN INCR RVN
; 3152      4678  3     FROM (IF ACTUALCOUNT () LSS 2 THEN 1 ELSE .P_RVN)
; 3153      4679  2     TO .SETCOUNT + .MTL[MTL_RVN_BASE] - 1
; 3154      4680  2 DO
; 3155      4681  3     BEGIN
; 3156      4682  3     CURRENT_VCB = VCB = .MTL[MTL_VCB(.RVN-.MTL[MTL_RVN_BASE])];
; 3157      4683  3     VCB[VCB_RVN] = .RVN;
; 3158      4684  3     CHANNEL = SWITCH_VOLUME (.RVN);
; 3159      4685  3
; 3160      4686  3     ! Get device characteristics of the device.
; 3161      4687  3     !
; 3162      4688  3     DESC[0] = DIB$C_LENGTH;
; 3163      4689  3     DESC[1] = DEVICE_CHAR;
; 3164      4690  3     STATUS = $GETCHN(CHAN=.CHANNEL, PRIBUF=DESC);
; 3165      4691  3     IF NOT .STATUS
; 3166      4692  3     THEN
; 3167      4693  3         SIGNAL(BACKUP$_GETCHN, 1, VCB[VCB_DEVICE], .STATUS);
; 3168      4694  3
; 3169      4695  3
; 3170      4696  3     IF .MODE NEQ 1 OR .QUAL[QUAL_COMP]
; 3171      4697  3     THEN
; 3172      4698  4         BEGIN
; 3173      4699  4         READ_HOMEBLOCK(.CHANNEL, DEVICE_CHAR, HOME_BLOCK);
; 3174      4700  4
; 3175      4701  4
; 3176      4702  4         ! Check the SETCOUNT and RVNs implied by the command against the
; 3177      4703  4         ! home block.
; 3178      4704  4         !
; 3179      4705  4         IF .RVN EQL 1
; 3180      4706  4         THEN
; 3181      4707  5             BEGIN
; 3182      4708  5             MTL[MTL_STRUCLEV] = .HOME_BLOCK[HM2$B_STRUCLEV];
; 3183      4709  5             IF .HOME_BLOCK[HM2$B_STRUCLEV] EQL 1
; 3184      4710  5             THEN
; 3185      4711  6                 BEGIN
; 3186      4712  6                 IF .SETCOUNT NEQ 1
; 3187      4713  6                 THEN
; 3188      4714  6                     SIGNAL(BACKUP$_INCSETCNT);
; 3189      4715  6                 END
; 3190      4716  5             ELSE
; 3191      4717  6                 BEGIN
; 3192      4718  6                 IF .MODE AND NOT .QUAL[QUAL_COMP]
; 3193      4719  6                 THEN
; 3194      4720  7                     BEGIN
; 3195      4721  7                     IF .HOME_BLOCK[HM2$W_SETCOUNT] NEQ 0
; 3196      4722  7                     THEN SIGNAL (BACKUP$_VOLINSET, 1, VCB[VCB_DEVICE]);
; 3197      4723  7                     END
; 3198      4724  6                 ELSE
; 3199      4725  7                     BEGIN
; 3200      4726  7                     IF .HOME_BLOCK[HM2$W_SETCOUNT] EQL 0
; 3201      4727  7                     AND .HOME_BLOCK[HM2$Q_RVN] NEQ 0
; 3202      4728  7                     THEN MTL[MTL_SEQ_DISK] = 1;
```

```
;  3203          4729  6                    END;
;  3204          4730  6
;  3205          4731  6                IF ACTUALCOUNT () LSS 2
;  3206          4732  6                OR NOT .MTL[MTL_SEQ_DISK]
;  3207          4733  6                THEN
;  3208          4734  7                    BEGIN
;  3209          4735  7                    IF
;  3210          4736  8                        BEGIN
;  3211          4737  8                        IF .SETCOUNT EQL 1
;  3212          4738  8                        THEN
;  3213          4739  8                            .HOME_BLOCK[HM2$W_RVN] NEQ 0
;  3214          4740  8                            AND .HOME_BLOCK[HM2$W_RVN] NEQ 1
;  3215          4741  8                        ELSE
;  3216          4742  8                            .HOME_BLOCK[HM2$W_RVN] NEQ .RVN
;  3217          4743  8                        END
;  3218          4744  7                    THEN
;  3219          4745  7                        SIGNAL(BACKUP$_INCRVN, 1, VCB[VCB_DEVICE]);
;  3220          4746  7
;  3221          4747  7
;  3222          4748  7                    IF
;  3223          4749  8                        BEGIN
;  3224          4750  8                        IF .SETCOUNT EQL 1
;  3225          4751  8                        THEN
;  3226          4752  8                            .HOME_BLOCK[HM2$W_SETCOUNT] NEQ 0
;  3227          4753  8                            AND .HOME_BLOCK[HM2$W_SETCOUNT] NEQ 1
;  3228          4754  8                        ELSE
;  3229          4755  8                            .HOME_BLOCK[HM2$W_SETCOUNT] NEQ .SETCOUNT
;  3230          4756  8                        END
;  3231          4757  7                    THEN
;  3232          4758  7                        SIGNAL(BACKUP$_INCSETCNT);
;  3233          4759  7                    END
;  3234          4760  7
;  3235          4761  6                ELSE
;  3236          4762  7                    BEGIN
;  3237          4763  7                    IF .HOME_BLOCK[HM2$W_RVN] NEQ 0
;  3238          4764  7                    THEN
;  3239          4765  8                        BEGIN
;  3240          4766  8                        RVN = .HOME_BLOCK[HM2$W_RVN];
;  3241          4767  8                        VCB[VCB_RVN] = .RVN;
;  3242          4768  8                        MTL[MTL_RVN_BASE] = .RVN;
;  3243          4769  8                        END
;  3244          4770  7                    ELSE
;  3245          4771  8                        BEGIN
;  3246          4772  8                        VCB[VCB_NOTVOLSET] = 1;
;  3247          4773  7                        END;
;  3248          4774  6                    END;
;  3249          4775  6
;  3250          4776  6                CH$MOVE(
;  3251          4777  6                    HM2$S_STRUCNAME,
;  3252          4778  6                    HOME_BLOCK[HM2$T_STRUCNAME],
;  3253          4779  6                    MTL[MTL_STRUCNAME]);
;  3254          4780  5                END;
;  3255          4781  5            END
;  3256          4782  4        ELSE
;  3257          4783  5            BEGIN
;  3258          4784  5            IF
;  3259          4785  5                .HOME_BLOCK[HM2$W_RVN] NEQ .RVN OR
```

```
; 3260          4786  5                 CH$NEQ(
; 3261          4787  5                     HM2$S_STRUCNAME, HOME_BLOCK[HM2$T_STRUCNAME],
; 3262          4788  5                     HM2$S_STRUCNAME, MTL[MTL_STRUCNAME])
; 3263          4789  5                 THEN
; 3264          4790  5                     SIGNAL(BACKUP$_INCRVN, 1, VCB[VCB_DEVICE]);
; 3265          4791  4                 END;
; 3266          4792  4
; 3267          4793  4
; 3268          4794  4             ! Finish initializing the VCB.
; 3269          4795  4             !
; 3270          4796  4             CH$MOVE (HM2$S_VOLNAME, HOME_BLOCK[HM2$T_VOLNAME], VCB[VCB_VOLNAME]);
; 3271          4797  4             IF .HOME_BLOCK[HM2$B_STRUCLEV] EQL 2
; 3272          4798  4             THEN
; 3273          4799  5                 BEGIN
; 3274          4800  5                 VCB[VCB_ODS_2] = TRUE;
; 3275          4801  5                 IF .HOME_BLOCK[HM2$V_NOHIGHWATER] THEN MTL[MTL_NOHWM] = TRUE;
; 3276          4802  5                 VCB[VCB_CLUSTER] = .HOME_BLOCK[HM2$W_CLUSTER];
; 3277          4803  5                 VCB[VCB_HDR_OFFSET] = .HOME_BLOCK[HM2$W_CLUSTER] * 4 + .HOME_BLOCK[HM2$W_IBMAPSIZE];
; 3278          4804  5                 VCB[VCB_MAXFILIDX] = .HOME_BLOCK[HM2$W_IBMAPSIZE] * 4096;
; 3279          4805  5                 VCB[VCB_IMAP_LBN] = .HOME_BLOCK[HM2$L_IBMAPLBN];
; 3280        P 4806  5                 STATUS = $QIOW(
; 3281        P 4807  5                     FUNC=IO$_READLBLK,
; 3282        P 4808  5                     CHAN=.CHANNEL,
; 3283        P 4809  5                     IOSB=IOSB,
; 3284        P 4810  5                     P1=HEADER,
; 3285        P 4811  5                     P2=512,
; 3286          4812  5                     P3=.HOME_BLOCK[HM2$L_IBMAPLBN] + .HOME_BLOCK[HM2$W_IBMAPSIZE]);
; 3287          4813  5                 IF .STATUS THEN STATUS = .IOSB[0];
; 3288          4814  5                 IF .STATUS THEN STATUS = VERIFY_HEADER(HEADER, INDEX_FILE_ID);
; 3289          4815  5                 IF NOT .STATUS
; 3290          4816  5                 THEN
; 3291          4817  6                     BEGIN
; 3292        P 4818  6                     STATUS = $QIOW(
; 3293        P 4819  6                         FUNC=IO$_READLBLK,
; 3294        P 4820  6                         CHAN=.CHANNEL,
; 3295        P 4821  6                         IOSB=IOSB,
; 3296        P 4822  6                         P1=HEADER,
; 3297        P 4823  6                         P2=512,
; 3298          4824  6                         P3=.HOME_BLOCK[HM2$L_ALTIDXLBN]);
; 3299          4825  6                     IF .STATUS THEN STATUS = .IOSB[0];
; 3300          4826  6                     IF .STATUS
; 3301          4827  6                     THEN
; 3302          4828  7                         BEGIN
; 3303          4829  7                         STATUS = VERIFY_HEADER(HEADER, INDEX_FILE_ID);
; 3304          4830  7                         IF NOT .STATUS THEN STATUS = SS$_BADFILEHDR;
; 3305          4831  6                         END;
; 3306          4832  5                     END;
; 3307          4833  5                 IF NOT .STATUS
; 3308          4834  5                 THEN
; 3309          4835  5                     SIGNAL(BACKUP$_NOINDEXF, 1, VCB[VCB_DEVICE], .STATUS);
; 3310          4836  5                 CREATE_WINDOW(HEADER, .RVN, VCB[VCB_INDEXF], 1, 0);
; 3311          4837  5
; 3312          4838  5                 ! If we are going to write on this disk, also read in the
; 3313          4839  5                 ! index file bitmap, and scan the storage map to build the
; 3314          4840  5                 ! allocation table.
; 3315          4841  5                 !
; 3316          4842  5                 IF .MODE AND NOT .QUAL[QUAL_COMP]
```

```
;  3317          4843  5              THEN
;  3318          4844  6                  BEGIN
;  3319          4845  6                  LOCAL
;  3320          4846  6                      BITS_SET,                   ! flag meaning ones seen
;  3321          4847  6                      EXTENT_COUNT,               ! count of extents found in bitmap
;  3322          4848  6                      FIRST_SET,                  ! starting LBN of extent
;  3323          4849  6                      LAST_SET;                   ! ending LBN of extent
;  3324          4850  6                  GLOBAL REGISTER
;  3325          4851  6                      COUNT=              6,      ! Retrieval pointer count
;  3326          4852  6                      LBN=                7,      ! Retrieval pointer LBN
;  3327          4853  6                      MAP_POINTER=        8: REF BBLOCK; ! Pointer to scan map area
;  3328          4854  6                  BIND
;  3329          4855  6                      BUFFER = HEADER : VECTOR;   ! buffer to read bitmap
;  3330          4856  6
;  3331          4857  6                  VCB[VCB_OUTPUT] = TRUE;
;  3332          4858  6                  VCB[VCB_SAVESET] = TRUE;
;  3333          4859  6                  VCB[VCB_INIT_DONE] = TRUE;
;  3334          4860  6                  VCB[VCB_IMAP] = GET_VM(.VCB[VCB_MAXFILIDX]/8);
;  3335        P 4861  6                  STATUS = $QIOW(
;  3336        P 4862  6                      FUNC=IO$_READLBLK,
;  3337        P 4863  6                      CHAN=.CHANNEL,
;  3338        P 4864  6                      IOSB=IOSB,
;  3339        P 4865  6                      P1=.VCB[VCB_IMAP],
;  3340        P 4866  6                      P2=.VCB[VCB_MAXFILIDX]/8,
;  3341          4867  6                      P3=.VCB[VCB_IMAP_LBN]);
;  3342          4868  6                  IF .STATUS THEN STATUS = .IOSB[0];
;  3343          4869  6                  IF NOT .STATUS
;  3344          4870  6                  THEN
;  3345          4871  6                      SIGNAL(BACKUP$_READIMAP, 1, VCB[VCB_DEVICE], .STATUS);
;  3346          4872  6
;  3347        P 4873  6                  STATUS = $QIOW(
;  3348        P 4874  6                      FUNC=IO$_READLBLK,
;  3349        P 4875  6                      CHAN=.CHANNEL,
;  3350        P 4876  6                      IOSB=IOSB,
;  3351        P 4877  6                      P1=HEADER,
;  3352        P 4878  6                      P2=512,
;  3353          4879  6                      P3=.VCB[VCB_IMAP_LBN] + .VCB[VCB_MAXFILIDX]/4096 + 1);
;  3354          4880  6                  IF .STATUS THEN STATUS = .IOSB[0];
;  3355          4881  6                  IF .STATUS
;  3356          4882  6                  THEN
;  3357          4883  7                      BEGIN
;  3358          4884  7                      STATUS = VERIFY_HEADER(HEADER, BITMAP_FILE_ID);
;  3359          4885  7                      IF NOT .STATUS THEN STATUS = SS$_BADFILEHDR;
;  3360          4886  6                      END;
;  3361          4887  6                  IF NOT .STATUS
;  3362          4888  6                  THEN
;  3363          4889  6                      SIGNAL(BACKUP$_NOBITMAP, 1, VCB[VCB_DEVICE], .STATUS);
;  3364          4890  6
;  3365          4891  6                  MAP_POINTER = HEADER + .HEADER[FH2$B_MPOFFSET]*2;
;  3366          4892  6                  GET_MAP_POINTER ();
;  3367          4893  6                  COUNT = .COUNT - 1;
;  3368          4894  9                  IF .COUNT NEQ (((.DEVICE_CHAR[DIB$L_MAXBLOCK]+.VCB[VCB_CLUSTER]-1)
;  3369          4895  7                                 / .VCB[VCB_CLUSTER] + 4095) / 4096)
;  3370          4896  6                  THEN SIGNAL (BACKUP$_NOBITMAP, 1, VCB[VCB_DEVICE]);
;  3371          4897  6                  VCB[VCB_BITMAP_LBN] = .LBN + 1;
;  3372          4898  6                  VCB[VCB_BITMAP_SIZE] = .COUNT;
;  3373          4899  6
```

```
; 3374          4900  6                   EXTENT_COUNT = 0;
; 3375          4901  6                   BITS_SET = FALSE;
; 3376          4902  6                   INCR VBN FROM 0 TO .COUNT-1
; 3377          4903  6                   DO
; 3378          4904  7                       BEGIN
; 3379        P 4905  7                       STATUS = $QIOW(
; 3380        P 4906  7                           FUNC=IO$_READLBLK,
; 3381        P 4907  7                           CHAN=.CHANNEL,
; 3382        P 4908  7                           IOSB=IOSB,
; 3383        P 4909  7                           P1=BUFFER,
; 3384        P 4910  7                           P2=512,
; 3385          4911  7                           P3=.VCB[VCB_BITMAP_LBN] + .VBN);
; 3386          4912  7                       IF .STATUS THEN STATUS = .IOSB[0];
; 3387          4913  7                       IF NOT .STATUS
; 3388          4914  7                       THEN
; 3389          4915  7                           SIGNAL(BACKUP$_READBMAP, 1, VCB[VCB_DEVICE], .STATUS);
; 3390          4916  7                       INCR J FROM 0 TO 127
; 3391          4917  7                       DO
; 3392          4918  8                           BEGIN
; 3393          4919  8                           INCR K FROM 0 TO 31
; 3394          4920  8                           DO
; 3395          4921  9                               BEGIN
; 3396          4922  9                               IF NOT .BITS_SET
; 3397          4923  9                               THEN
; 3398          4924 10                                   BEGIN
; 3399          4925 10                                   IF .BUFFER[.J] EQL 0 THEN EXITLOOP;
; 3400          4926 10                                   IF .BITVECTOR [BUFFER[.J], .K]
; 3401          4927 10                                   THEN
; 3402          4928 11                                       BEGIN
; 3403          4929 11                                       BITS_SET = TRUE;
; 3404          4930 11                                       FIRST_SET = (.VBN*4096 + .J*32 + .K) * .VCB[VCB_CLUSTER];
; 3405          4931 10                                       END;
; 3406          4932 10                                   END
; 3407          4933  9                               ELSE
; 3408          4934 10                                   BEGIN
; 3409          4935 10                                   IF .BUFFER[.J] EQL -1 THEN EXITLOOP;
; 3410          4936 10                                   IF NOT .BITVECTOR [BUFFER[.J], .K]
; 3411          4937 10                                   THEN
; 3412          4938 11                                       BEGIN
; 3413          4939 11                                       BITS_SET = FALSE;
; 3414          4940 11                                       LAST_SET = (.VBN*4096 + .J*32 + .K) * .VCB[VCB_CLUSTER];
; 3415          4941 11                                       FREE_BLOCKS (.LAST_SET-.FIRST_SET, .FIRST_SET);
; 3416          4942 11                                       EXTENT_COUNT = .EXTENT_COUNT + 1;
; 3417          4943 11                                       IF .EXTENT_COUNT GTR 100
; 3418          4944 11                                       THEN SIGNAL (BACKUP$_DISKFRAG, 1, VCB[VCB_DEVICE]);
; 3419          4945 10                                       END;
; 3420          4946  9                                   END;
; 3421          4947  8                               END;
; 3422          4948  7                           END;
; 3423          4949  6                       END;
; 3424          4950  6                   IF .BITS_SET
; 3425          4951  6                   THEN FREE_BLOCKS (.COUNT*4096*.VCB[VCB_CLUSTER]-.FIRST_SET, .FIRST_SET);
; 3426          4952  5                   END;
; 3427          4953  5               END
; 3428          4954  5
; 3429          4955  5           ! Finish setup of ODS-1 disk. Save set processing is not supported.
; 3430          4956  5           !
```

```
;3431          4957  4          ELSE
;3432          4958  5              BEGIN
;3433          4959  5              IF .MODE
;3434          4960  5              THEN SIGNAL (BACKUP$_ODS2SAVE, 1, VCB[VCB_DEVICE]);
;3435          4961  5              VCB[VCB_CLUSTER] = 1;
;3436          4962  5              VCB[VCB_HDR_OFFSET] = 2 + .HOME_BLOCK[HM1$W_IBMAPSIZE];
;3437          4963  5              VCB[VCB_MAXFILIDX] = .HOME_BLOCK[HM1$W_IBMAPSIZE] * 4096;
;3438        P 4964  5              STATUS = $QIOW(
;3439        P 4965  5                  FUNC=IO$_READLBLK,
;3440        P 4966  5                  CHAN=.CHANNEL,
;3441        P 4967  5                  IOSB=IOSB,
;3442        P 4968  5                  P1=HEADER,
;3443        P 4969  5                  P2=512,
;3444          4970  5                  P3=ROT(.HOME_BLOCK[HM1$L_IBMAPLBN], 16) + .HOME_BLOCK[HM1$W_IBMAPSIZE]);
;3445          4971  5              IF .STATUS THEN STATUS = .IOSB[0];
;3446          4972  5              IF .STATUS
;3447          4973  5              THEN
;3448          4974  6                  BEGIN
;3449          4975  6                  STATUS = VERIFY_HEADER(HEADER, INDEX_FILE_ID);
;3450          4976  6                  IF NOT .STATUS THEN STATUS = SS$_BADFILEHDR;
;3451          4977  5                  END;
;3452          4978  5              IF NOT .STATUS
;3453          4979  5              THEN
;3454          4980  5                  SIGNAL(BACKUP$_NOINDEXF, 1, VCB[VCB_DEVICE], .STATUS);
;3455          4981  5              CREATE_WINDOW(HEADER, 1, VCB[VCB_INDEXF], 1, 0);
;3456          4982  4              END;
;3457          4983  3          END;
;3458          4984  3      IF ACTUALCOUNT () GEQ 2
;3459          4985  3      AND .MTL[MTL_SEQ_DISK] THEN EXITLOOP;
;3460          4986  2      END;
;3461          4987  1 END;


                0000  0001  0001  01318 P.AAG:  .WORD   1, 1, 0                                                    ;
                0000  0002  0002  0131E P.AAH:  .WORD   2, 2, 0                                                    ;

                                        INDEX_FILE_ID=      P.AAG
                                        BITMAP_FILE_ID=     P.AAH


                                  0FFC 00000            .ENTRY  STA_MOUNT, Save R2,R3,R4,R5,R6,R7,R8,R9,-        ;4559
                                                                R10,R11
                5E      FB68      CE   9E 00002                 MOVAB   -1176(SP), SP                             ;
                08        04      AC   E9 00007                 BLBC    MODE, 1$                                  ;4621
                    00000000'     EF   D5 0000B                 TSTL    OUTPUT_MTL                                ;
                                  06   11 00011                 BRB     2$                                        ;
                    00000000'     EF   D5 00013 1$:             TSTL    INPUT_MTL                                 ;
                                  03   13 00019 2$:             BEQL    3$                                        ;
                                00DA   31 0001B                 BRW     15$                                       ;
                09        04      AC   E9 0001E 3$:             BLBC    MODE, 4$                                  ;4624
                52  00000000'     EF   D0 00022                 MOVL    QUAL+4, Q                                 ;
                                  07   11 00029                 BRB     5$                                        ;
                52  00000000'     EF   D0 0002B 4$:             MOVL    QUAL, Q                                   ;
                                  59   D4 00032 5$:             CLRL    SETCOUNT                                  ;4625
                                  52   D5 00034 6$:             TSTL    Q                                         ;4626
                                  07   13 00036                 BEQL    7$                                        ;
```

```
                                 59  D6 00038          INCL    SETCOUNT                        ; 4628
                              52 62  D0 0003A          MOVL    (Q), Q                          ; 4629
                                 F5  11 0003D          BRB     6$                              ; 4626
                        7E       59  02 78 0003F 7$:    ASHL    #2, SETCOUNT, -(SP)             ; 4635
                           6E    34  C0 00043          ADDL2   #52, (SP)                       ;
                00000000G  00    01  FB 00046          CALLS   #1, GET_ZERO_VM                 ;
                           6E    50  D0 0004D          MOVL    R0, MTL                         ;
                00000000'  EF    6E  D0 00050          MOVL    MTL, CURRENT_MTL                ; 4636
                           15 04 AC  E9 00057          BLBC    MODE, 8$                        ; 4637
                00000000'  EF    6E  D0 0005B          MOVL    MTL, OUTPUT_MTL                 ; 4640
                           02    6C  91 00062          CMPB    (AP), #2                        ; 4641
                                 10  1F 00065          BLSSU   9$                              ;
                     50    6E    31  C1 00067          ADDL3   #49, MTL, R0                    ;
                           60    01  88 0006B          BISB2   #1, (R0)                        ;
                                 07  11 0006E          BRB     9$                              ; 4637
                00000000'  EF    6E  D0 00070 8$:     MOVL    MTL, INPUT_MTL                  ; 4644
                     50    6E    1F  C1 00077 9$:     ADDL3   #31, MTL, R0                    ; 4645
                           60    59  90 0007B          MOVB    SETCOUNT, (R0)                  ;
                     50    6E    30  C1 0007E          ADDL3   #48, MTL, R0                    ; 4646
                           60    01  90 00082          MOVB    #1, (R0)                        ;
                           7E 0200 8F  3C 00085          MOVZWL  #512, -(SP)                     ; 4647
                00000000G  00    01  FB 0008A          CALLS   #1, GET_VM                      ;
                     53    6E    0C  C1 00091          ADDL3   #12, MTL, R3                    ;
                           63    50  D0 00095          MOVL    R0, (R3)                        ;
                           09 04 AC  E9 00098          BLBC    MODE, 10$                       ; 4652
                           52 00000000' EF  D0 0009C    MOVL    QUAL+4, Q                       ;
                                 07  11 000A3          BRB     11$                             ;
                           52 00000000' EF  D0 000A5 10$:    MOVL    QUAL, Q                         ;
                           54    59  D0 000AC 11$:    MOVL    SETCOUNT, R4                    ; 4653
                                 53  D4 000AF          CLRL    RVN                             ;
                                 41  11 000B1          BRB     14$                             ;
                           7E    44  8F  9A 000B3 12$:    MOVZBL  #68, -(SP)                      ; 4655
                00000000G  00    01  FB 000B7          CALLS   #1, GET_ZERO_VM                 ;
                           5B    50  D0 000BE          MOVL    R0, VCB                         ;
                     55    6E    30  C1 000C1          ADDL3   #48, MTL, R5                    ;
                        6543       5B  D0 000C5          MOVL    VCB, (R5)[RVN]                  ;
                     06    AB    53  90 000C9          MOVB    RVN, 6(VCB)                     ; 4656
                           50 28 AB  9E 000CD          MOVAB   40(VCB), R0                     ; 4657
                        2C    AB    50  D0 000D1          MOVL    R0, 44(VCB)                     ;
                        28    AB    50  D0 000D5          MOVL    R0, 40(VCB)                     ;
                              04  AC  D5 000D9          TSTL    MODE                            ; 4658
                                 04  13 000DC          BEQL    13$                             ;
                        07    AB    01  88 000DE          BISB2   #1, 7(VCB)                      ;
                        30    AB 04 A2  D0 000E2 13$:    MOVL    4(Q), 48(VCB)                   ; 4659
                        20    AB 10 A2  B0 000E7          MOVW    16(Q), 32(VCB)                  ; 4661
                        24    AB 14 A2  D0 000EC          MOVL    20(Q), 36(VCB)                  ; 4663
                              52  62  D0 000F1          MOVL    (Q), Q                          ; 4664
                     BB       53  F3 000F4 14$:    AOBLEQ  R4, RVN, 12$                    ; 4653
                           09 04 AC  E9 000F8 15$:    BLBC    MODE, 16$                       ; 4671
                           6E 00000000' EF  D0 000FC    MOVL    OUTPUT_MTL, MTL                 ; 4672
                                 07  11 00103          BRB     17$                             ;
                           6E 00000000' EF  D0 00105 16$:    MOVL    INPUT_MTL, MTL                  ; 4673
                00000000'  EF    6E  D0 0010C 17$:    MOVL    MTL, CURRENT_MTL                ;
                     50    6E    1F  C1 00113          ADDL3   #31, MTL, R0                    ; 4674
                           59    60  9A 00117          MOVZBL  (R0), SETCOUNT                  ;
                           02    6C  91 0011A          CMPB    (AP), #2                        ; 4675
                                 06  1F 0011D          BLSSU   18$                             ;
```

```
                                    08    AC  D5 0011F             TSTL    P_RVN                                   ; 4676
                                          01  12 00122             BNEQ    18$                                     ;
                                              04 00124             RET                                             ;
                              02            6C  91 00125 18$:      CMPB    (AP), #2                                ; 4678
                                          05  1E 00128             BGEQU   19$                                     ;
                              5A            01  D0 0012A           MOVL    #1, R10                                 ;
                                          04  11 0012D             BRB     20$                                     ;
                              5A    08      AC  D0 0012F 19$:      MOVL    P_RVN, R10                              ;
                    51        6E          30  C1 00133 20$:      ADDL3   #48, MTL, R1                            ; 4679
                              50          61  9A 00137             MOVZBL  (R1), R0                                ;
                10  AE        59          50  C1 0013A             ADDL3   R0, SETCOUNT, 16(SP)                    ;
                                          5A  D7 0013F             DECL    RVN                                     ; 4677
                                      058C  31 00141               BRW     70$                                     ;
                    51        6E          30  C1 00144 21$:      ADDL3   #48, MTL, R1                            ; 4682
                              50          61  9A 00148             MOVZBL  (R1), R0                                ;
                    50        5A          50  C3 0014B             SUBL3   R0, RVN, R0                             ;
                    52        6E          34  C1 0014F             ADDL3   #52, MTL, R2                            ;
                              5B        6240  D0 00153             MOVL    (R2)[R0], VCB                           ;
                00000000'     EF          5B  D0 00157             MOVL    VCB, CURRENT_VCB                        ;
                        06    AB          5A  90 0015E             MOVB    RVN, 6(VCB)                             ; 4683
                                          5A  DD 00162             PUSHL   RVN                                     ; 4684
                    ECB4      CF          01  FB 00164             CALLS   #1, SWITCH_VOLUME                       ;
                      08      AE          50  D0 00169             MOVL    R0, CHANNEL                             ;
                    FF7C      CD    74    8F  9A 0016D             MOVZBL  #116, DESC                              ; 4688
                      80      AD    84    AD  9E 00173             MOVAB   DEVICE_CHAR, DESC+4                     ; 4689
                                          7E  7C 00178             CLRQ    -(SP)                                   ; 4690
                                    FF7C  CD  9F 0017A             PUSHAB  DESC                                    ;
                                          7E  D4 0017E             CLRL    -(SP)                                   ;
                                    18    AE  DD 00180             PUSHL   CHANNEL                                 ;
            00000000G   00                  05  FB 00183           CALLS   #5, SYS$GETCHN                          ;
                  04    AE                  50  D0 0018A           MOVL    R0, STATUS                              ;
                        15          04    AE  E8 0018E             BLBS    STATUS, 22$                             ; 4691
                                    04    AE  DD 00192             PUSHL   STATUS                                  ; 4693
                                    20    AB  9F 00195             PUSHAB  32(VCB)                                 ;
                                          01  DD 00198             PUSHL   #1                                      ;
                                00000000G   8F  DD 0019A           PUSHL   #BACKUP$_GETCHN                         ;
            00000000G   00                  04  FB 001A0           CALLS   #4, LIB$SIGNAL                          ;
                        01          04    AC  D1 001A7 22$:      CMPL    MODE, #1                                ; 4696
                                          0B  12 001AB             BNEQ    23$                                     ;
                                00000000'   EF  95 001AD           TSTB    QUAL+8                                  ;
                                          03  19 001B3             BLSS    23$                                     ;
                                      050C  31 001B5               BRW     69$                                     ;
                                    0214    CE  9F 001B8 23$:      PUSHAB  HOME_BLOCK                              ; 4699
                                      84    AD  9F 001BC           PUSHAB  DEVICE_CHAR                             ;
                                      10    AE  DD 001BF           PUSHL   CHANNEL                                 ;
                          F833      CF          03  FB 001C2           CALLS   #3, READ_HOMEBLOCK                      ;
                                    01          5A  D1 001C7           CMPL    RVN, #1                                 ; 4705
                                          03  13 001CA             BEQL    24$                                     ;
                                      00E0  31 001CC               BRW     38$                                     ;
            50                    6E          1E  C1 001CF 24$:      ADDL3   #30, MTL, R0                            ; 4708
                                    60    0221  CE  90 001D3           MOVB    HOME_BLOCK+13, (R0)                     ;
                                    01    0221  CE  91 001D8           CMPB    HOME_BLOCK+13, #1                       ; 4709
                                          15  12 001DD             BNEQ    26$                                     ;
                                    01          59  D1 001DF           CMPL    SETCOUNT, #1                            ; 4712
                                          0D  13 001E2             BEQL    25$                                     ;
                                00000000G   8F  DD 001E4           PUSHL   #BACKUP$_INCSETCNT                      ; 4714
            00000000G   00                  01  FB 001EA           CALLS   #1, LIB$SIGNAL                          ;
```

```
                                                00E2  31 001F1 25$:     BRW     40$                                     ; 4709
                                          50     023C  CE  3C 001F4 26$:     MOVZWL  HOME_BLOCK+40, R0                       ; 4721
                                          20       04  AC  E9 001F9         BLBC    MODE, 27$                               ; 4718
                                             00000000'  EF  95 001FD         TSTB    QUAL+8                                  ;
                                                        18  19 00203         BLSS    27$                                     ;
                                                        50  D5 00205         TSTL    R0                                      ; 4721
                                                        25  13 00207         BEQL    28$                                     ;
                                                    20      AB  9F 00209         PUSHAB  32(VCB)                             ; 4722
                                                        01  DD 0020C         PUSHL   #1                                      ;
                                                 00000000G  8F  DD 0020E         PUSHL   #BACKUP$_VOLINSET                       ;
                                      00000000G  00         03  FB 00214         CALLS   #3, LIB$SIGNAL                      ;
                                                        11  11 0021B         BRB     28$                                     ; 4718
                                                        50  D5 0021D 27$:     TSTL    R0                                      ; 4726
                                                        0D  12 0021F         BNEQ    28$                                     ;
                                                    023A    CE  B5 00221         TSTW    HOME_BLOCK+38                           ; 4727
                                                        07  13 00225         BEQL    28$                                     ;
                                          50         6E     31  C1 00227         ADDL3   #49, MTL, R0                        ; 4728
                                          60                01  88 0022B         BISB2   #1, (R0)                            ;
                                          50     023A    CE  3C 0022E 28$:     MOVZWL  HOME_BLOCK+38, R0                   ; 4739
                                          02                6C  91 00233         CMPB    (AP), #2                            ; 4731
                                                        07  1F 00236         BLSSU   29$                                     ;
                                          51         6E     31  C1 00238         ADDL3   #49, MTL, R1                        ; 4732
                                          4C                61  E8 0023C         BLBS    (R1), 35$                           ;
                                          01                59  D1 0023F 29$:     CMPL    SETCOUNT, #1                        ; 4737
                                                        09  12 00242         BNEQ    30$                                     ;
                                                        50  D5 00244         TSTL    R0                                      ; 4739
                                                        1C  13 00246         BEQL    32$                                     ;
                                          01                50  B1 00248         CMPW    R0, #1                              ; 4740
                                                        03  11 0024B         BRB     31$                                     ;
                                          5A                50  D1 0024D 30$:     CMPL    R0, RVN                             ; 4742
                                                        12  13 00250 31$:     BEQL    32$                                     ;
                                                    20      AB  9F 00252         PUSHAB  32(VCB)                             ; 4745
                                                        01  DD 00255         PUSHL   #1                                      ;
                                                 00000000G  8F  DD 00257         PUSHL   #BACKUP$_INCRVN                         ;
                                      00000000G  00         03  FB 0025D         CALLS   #3, LIB$SIGNAL                      ;
                                          50     023C    CE  3C 00264 32$:     MOVZWL  HOME_BLOCK+40, R0                   ; 4752
                                          01                59  D1 00269         CMPL    SETCOUNT, #1                        ; 4750
                                                        09  12 0026C         BNEQ    33$                                     ;
                                                        50  D5 0026E         TSTL    R0                                      ; 4752
                                                        31  13 00270         BEQL    37$                                     ;
                                          01                50  B1 00272         CMPW    R0, #1                              ; 4753
                                                        03  11 00275         BRB     34$                                     ;
                                          59                50  D1 00277 33$:     CMPL    R0, SETCOUNT                        ; 4755
                                                        27  13 0027A 34$:     BEQL    37$                                     ;
                                                 00000000G  8F  DD 0027C         PUSHL   #BACKUP$_INCSETCNT                  ; 4758
                                      00000000G  00         01  FB 00282         CALLS   #1, LIB$SIGNAL                      ;
                                                        18  11 00289         BRB     37$                                     ; 4731
                                                        50  D5 0028B 35$:     TSTL    R0                                      ; 4763
                                                        10  13 0028D         BEQL    36$                                     ;
                                          5A                50  D0 0028F         MOVL    R0, RVN                             ; 4766
                                             06     AB     5A  90 00292         MOVB    RVN, 6(VCB)                         ; 4767
                                          50         6E     30  C1 00296         ADDL3   #48, MTL, R0                        ; 4768
                                          60                5A  90 0029A         MOVB    RVN, (R0)                           ;
                                                        04  11 0029D         BRB     37$                                     ; 4763
                                             07     AB     10  88 0029F 36$:     BISB2   #16, 7(VCB)                         ; 4772
                                          56         6E     24  C1 002A3 37$:     ADDL3   #36, MTL, R6                        ; 4779
                                          66     FF48    CD  0C  28 002A7         MOVC3   #12, HOME_BLOCK+460, (R6)           ;
```

```
                                                  27  11 002AD            BRB     40$                                     ; 4705
          5A      023A    CE              10      00  ED 002AF 38$:       CMPZV   #0, #16, HOME_BLOCK+38, RVN            ; 4785
                                                  0C  12 002B6            BNEQ    39$                                     ;
                  54                      6E      24  C1 002B8            ADDL3   #36, MTL, R4                            ; 4788
                  64      FF48    CD              0C  29 002BC            CMPC3   #12, HOME_BLOCK+460, (R4)               ;
                                                  12  13 002C2            BEQL    40$                                     ;
                                              20  AB  9F 002C4 39$:       PUSHAB  32(VCB)                                 ; 4790
                                                  01  DD 002C7            PUSHL   #1                                      ;
                                      00000000G   8F  DD 002C9            PUSHL   #BACKUP$_INCRVN                         ;
                          00000000G   00          03  FB 002CF            CALLS   #3, LIB$SIGNAL                          ;
                  38  AB      FF54    CD          0C  28 002D6 40$:       MOVC3   #12, HOME_BLOCK+472, 56(VCB)            ; 4796
                                      02  0221    CE  91 002DD            CMPB    HOME_BLOCK+13, #2                       ; 4797
                                                  03  13 002E2            BEQL    41$                                     ;
                                                  0332  31 002E4          BRW     65$                                     ;
                                          07  AB      02  88 002E7 41$:   BISB2   #2, 7(VCB)                              ; 4800
                                  07      023E    CE  03  E1 002EB        BBC     #3, HOME_BLOCK+42, 42$                  ; 4801
                                  50              6E  31  C1 002F1        ADDL3   #49, MTL, R0                            ;
                                                  60  04  88 002F5        BISB2   #4, (R0)                                ;
                                          04  AB      0222    CE  B0 002F8 42$:   MOVW    HOME_BLOCK+14, 4(VCB)           ; 4802
                                                  50  0222    CE  3C 002FE    MOVZWL  HOME_BLOCK+14, R0                   ; 4803
                                                  51  0234    CE  3C 00303    MOVZWL  HOME_BLOCK+32, R1                   ;
                                                  52      6140  DE 00308      MOVAL   (R1)[R0], R2                        ;
                                          1A  AB          52  B0 0030C        MOVW    R2, 26(VCB)                         ;
                                                  50  0234    CE  3C 00310    MOVZWL  HOME_BLOCK+32, R0                   ; 4804
                                  1C  AB          50  0C  78 00315            ASHL    #12, R0, 28(VCB)                    ;
                                          14  AB      022C    CE  D0 0031A    MOVL    HOME_BLOCK+24, 20(VCB)              ; 4805
                                                      7E  7C 00320            CLRQ    -(SP)                               ; 4812
                                                      7E  D4 00322            CLRL    -(SP)                               ;
                                                  50  0240    CE  3C 00324    MOVZWL  HOME_BLOCK+32, R0                   ;
                                                  0238  DE40  9F 00329        PUSHAB  @HOME_BLOCK+24[R0]                  ;
                                                  7E  0200    8F  3C 0032E    MOVZWL  #512, -(SP)                         ;
                                                      28  AE  9F 00333        PUSHAB  HEADER                              ;
                                                      7E  7C 00336            CLRQ    -(SP)                               ;
                                                      F8  AD  9F 00338        PUSHAB  IOSB                                ;
                                                      21  DD 0033B            PUSHL   #33                                 ;
                                                      30  AE  DD 0033D        PUSHL   CHANNEL                             ;
                                                      7E  D4 00340            CLRL    -(SP)                               ;
                          00000000G   00              0C  FB 00342            CALLS   #12, SYS$QIOW                       ;
                                  04  AE              50  D0 00349            MOVL    R0, STATUS                          ;
                                      1D      04  AE  E9 0034D            BLBC    STATUS, 43$                             ; 4813
                                  04  AE      F8  AD  3C 00351            MOVZWL  IOSB, STATUS                            ;
                                      14      04  AE  E9 00356            BLBC    STATUS, 43$                             ; 4814
                                          FC96    CF  9F 0035A            PUSHAB  INDEX_FILE_ID                           ;
                                              18  AE  9F 0035E            PUSHAB  HEADER                                  ;
                                  EB50    CF          02  FB 00361        CALLS   #2, VERIFY_HEADER                       ;
                                  04  AE              50  D0 00366        MOVL    R0, STATUS                              ;
                                      67      04  AE  E8 0036A            BLBS    STATUS, 45$                             ; 4815
                                                      7E  7C 0036E 43$:   CLRQ    -(SP)                               ; 4824
                                                      7E  D4 00370        CLRL    -(SP)                               ;
                                                  0228    CE  DD 00372    PUSHL   HOME_BLOCK+8                            ;
                                      7E          0200    8F  3C 00376    MOVZWL  #512, -(SP)                             ;
                                                      28  AE  9F 0037B    PUSHAB  HEADER                                  ;
                                                      7E  7C 0037E        CLRQ    -(SP)                                   ;
                                                      F8  AD  9F 00380    PUSHAB  IOSB                                    ;
                                                      21  DD 00383        PUSHL   #33                                     ;
                                                      30  AE  DD 00385    PUSHL   CHANNEL                                 ;
                                                      7E  D4 00388        CLRL    -(SP)                                   ;
```

```
                00000000G  00              0C  FB 0038A          CALLS   #12, SYS$QIOW                                    ;
                      04   AE              50  D0 00391          MOVL    R0, STATUS                                      ;
                           27        04    AE  E9 00395          BLBC    STATUS, 44$                                     ; 4825
                      04   AE        F8    AD  3C 00399          MOVZWL  IOSB, STATUS                                    ;
                           1E        04    AE  E9 0039E          BLBC    STATUS, 44$                                     ; 4826
                                   FC4E    CF  9F 003A2          PUSHAB  INDEX_FILE_ID                                   ; 4829
                                     18    AE  9F 003A6          PUSHAB  HEADER                                          ;
                    EB08   CF              02  FB 003A9          CALLS   #2, VERIFY_HEADER                               ;
                      04   AE              50  D0 003AE          MOVL    R0, STATUS                                      ;
                           1F        04    AE  E8 003B2          BLBS    STATUS, 45$                                     ; 4830
                      04   AE      0810    8F  3C 003B6          MOVZWL  #2064, STATUS                                   ;
                           15        04    AE  E8 003BC          BLBS    STATUS, 45$                                     ; 4833
                                     04    AE  DD 003C0 44$:     PUSHL   STATUS                                          ; 4835
                                     20    AB  9F 003C3          PUSHAB  32(VCB)                                         ;
                                             01  DD 003C6          PUSHL   #1                                              ;
                                00000000G    8F  DD 003C8          PUSHL   #BACKUP$_NOINDEXF                               ;
                00000000G  00              04  FB 003CE          CALLS   #4, LIB$SIGNAL                                  ;
                           7E              01  7D 003D5 45$:     MOVQ    #1, -(SP)                                       ; 4836
                           7E              5A  7D 003D8          MOVQ    RVN, -(SP)                                      ;
                                     24    AE  9F 003DB          PUSHAB  HEADER                                          ;
                    F0C7   CF              05  FB 003DE          CALLS   #5, CREATE_WINDOW                               ;
                           03        04    AC  E8 003E3          BLBS    MODE, 47$                                       ; 4842
                                         02DA  31 003E7 46$:     BRW     69$                                             ;
                                00000000'    EF  95 003EA 47$:     TSTB    QUAL+8                                          ;
                                             F5  19 003F0          BLSS    46$                                             ;
                      07   AB              0D  88 003F2          BISB2   #13, 7(VCB)                                     ; 4859
                7E    1C   AB              08  C7 003F6          DIVL3   #8, 28(VCB), -(SP)                              ; 4860
                00000000G  00              01  FB 003FB          CALLS   #1, GET_VM                                      ;
                      10   AB              50  D0 00402          MOVL    R0, 16(VCB)                                     ;
                                             7E  7C 00406          CLRQ    -(SP)                                           ; 4867
                                             7E  D4 00408          CLRL    -(SP)                                           ;
                                     14    AB  DD 0040A          PUSHL   20(VCB)                                         ;
                7E    1C   AB                08  C7 0040D          DIVL3   #8, 28(VCB), -(SP)                              ;
                                     10    AB  DD 00412          PUSHL   16(VCB)                                         ;
                                             7E  7C 00415          CLRQ    -(SP)                                           ;
                                     F8    AD  9F 00417          PUSHAB  IOSB                                            ;
                                             21  DD 0041A          PUSHL   #33                                             ;
                                     30    AE  DD 0041C          PUSHL   CHANNEL                                         ;
                                             7E  D4 0041F          CLRL    -(SP)                                           ;
                00000000G  00              0C  FB 00421          CALLS   #12, SYS$QIOW                                   ;
                      04   AE              50  D0 00428          MOVL    R0, STATUS                                      ;
                           09        04    AE  E9 0042C          BLBC    STATUS, 48$                                     ; 4868
                      04   AE        F8    AD  3C 00430          MOVZWL  IOSB, STATUS                                    ;
                           15        04    AE  E8 00435          BLBS    STATUS, 49$                                     ; 4869
                                     04    AE  DD 00439 48$:     PUSHL   STATUS                                          ; 4871
                                     20    AB  9F 0043C          PUSHAB  32(VCB)                                         ;
                                             01  DD 0043F          PUSHL   #1                                              ;
                                00000000G    8F  DD 00441          PUSHL   #BACKUP$_READIMAP                               ;
                00000000G  00              04  FB 00447          CALLS   #4, LIB$SIGNAL                                  ;
                                             7E  7C 0044E 49$:     CLRQ    -(SP)                                           ; 4879
                                             7E  D4 00450          CLRL    -(SP)                                           ;
                50    1C   AB  00001000    8F  C7 00452          DIVL3   #4096, 28(VCB), R0                              ;
                           50        14    AB  C0 0045B          ADDL2   20(VCB), R0                                     ;
                                     01    A0  9F 0045F          PUSHAB  1(R0)                                           ;
                           7E      0200    8F  3C 00462          MOVZWL  #512, -(SP)                                     ;
                                     28    AE  9F 00467          PUSHAB  HEADER                                          ;
                                             7E  7C 0046A          CLRQ    -(SP)                                           ;
```

```
                                F8    AD   9F 0046C          PUSHAB  IOSB                                    ;
                                21    DD   0046F          PUSHL   #33                                     ;
                                30    AE   DD 00471          PUSHL   CHANNEL                                 ;
                                      7E   D4 00474          CLRL    -(SP)                                   ;
                00000000G       00          0C   FB 00476          CALLS   #12, SYS$QIOW                   ;
                        04      AE          50   D0 0047D          MOVL    R0, STATUS                      ;
                                27      04  AE   E9 00481          BLBC    STATUS, 50$                     ; 4880
                        04      AE      F8  AD   3C 00485          MOVZWL  IOSB, STATUS                    ;
                                1E      04  AE   E9 0048A          BLBC    STATUS, 50$                     ; 4881
                                      FB68  CF   9F 0048E          PUSHAB  BITMAP_FILE_ID                  ; 4884
                                        18  AE   9F 00492          PUSHAB  HEADER                          ;
                        EA1C    CF          02   FB 00495          CALLS   #2, VERIFY_HEADER               ;
                        04      AE          50   D0 0049A          MOVL    R0, STATUS                      ;
                                1F      04  AE   E8 0049E          BLBS    STATUS, 51$                     ; 4885
                        04      AE    0810  8F   3C 004A2          MOVZWL  #2064, STATUS                   ;
                                15      04  AE   E8 004A8          BLBS    STATUS, 51$                     ; 4887
                                        04  AE   DD 004AC  50$:     PUSHL   STATUS                          ; 4889
                                        20  AB   9F 004AF          PUSHAB  32(VCB)                         ;
                                            01   DD 004B2          PUSHL   #1                              ;
                                00000000G   8F   DD 004B4          PUSHL   #BACKUP$_NOBITMAP               ;
                00000000G       00          04   FB 004BA          CALLS   #4, LIB$SIGNAL                  ;
                                50      15  AE   9A 004C1  51$:     MOVZBL  HEADER+1, R0                    ; 4891
                                58      14  AE40 3E 004C5          MOVAW   HEADER[R0], MAP_POINTER         ;
                                00000000G   00   16 004CA          JSB     GET_MAP_POINTER                 ; 4892
                                            56   D7 004D0          DECL    COUNT                           ; 4893
                                50      04  AB   3C 004D2          MOVZWL  4(VCB), R0                      ; 4894
                                50      F4  AD   C0 004D6          ADDL2   DEVICE_CHAR+112, R0             ;
                                            50   D7 004DA          DECL    R0                              ;
                                51      04  AB   3C 004DC          MOVZWL  4(VCB), R1                      ; 4895
                                50          51   C6 004E0          DIVL2   R1, R0                          ;
                                50    0FFF  C0   9E 004E3          MOVAB   4095(R0), R0                    ; 4894
                                50 00001000 8F   C6 004E8          DIVL2   #4096, R0                       ; 4895
                                50          56   D1 004EF          CMPL    COUNT, R0                       ; 4894
                                            12   13 004F2          BEQL    52$                             ;
                                        20  AB   9F 004F4          PUSHAB  32(VCB)                         ; 4896
                                            01   DD 004F7          PUSHL   #1                              ;
                                00000000G   8F   DD 004F9          PUSHL   #BACKUP$_NOBITMAP               ;
                00000000G       00          03   FB 004FF          CALLS   #3, LIB$SIGNAL                  ;
                        0C      AB      01  A7   9E 00506  52$:     MOVAB   1(R7), 12(VCB)                  ; 4897
                        34      AB          56   B0 0050B          MOVW    COUNT, 52(VCB)                  ; 4898
                                            57   7C 0050F          CLRQ    BITS_SET                        ; 4901
                                53          01   CE 00511          MNEGL   #1, VBN                         ; 4902
                                          0DD   31   00514          BRW     61$                             ;
                                            7E   7C 00517  53$:     CLRQ    -(SP)                           ; 4911
                                            7E   D4 00519          CLRL    -(SP)                           ;
                                        0C  BB43 9F 0051B          PUSHAB  @12(VCB)[VBN]                   ;
                                7E      0200  8F   3C 0051F          MOVZWL  #512, -(SP)                     ;
                                        28  AE   9F 00524          PUSHAB  BUFFER                          ;
                                            7E   7C 00527          CLRQ    -(SP)                           ;
                                        F8  AD   9F 00529          PUSHAB  IOSB                            ;
                                            21   DD 0052C          PUSHL   #33                             ;
                                        30  AE   DD 0052E          PUSHL   CHANNEL                         ;
                                            7E   D4 00531          CLRL    -(SP)                           ;
                00000000G       00          0C   FB 00533          CALLS   #12, SYS$QIOW                   ;
                        04      AE          50   D0 0053A          MOVL    R0, STATUS                      ;
                                09      04  AE   E9 0053E          BLBC    STATUS, 54$                     ; 4912
                        04      AE      F8  AD   3C 00542          MOVZWL  IOSB, STATUS                    ;
```

```
                                15      04   AE   E8 00547              BLBS    STATUS, 55$                                     ; 4913
                                        04   AE   DD 0054B 54$:         PUSHL   STATUS                                          ; 4915
                                        20   AB   9F 0054E              PUSHAB  32(VCB)                                         ;
                                        01   DD   00551                PUSHL   #1                                              ;
                                 00000000G  8F   DD 00553              PUSHL   #BACKUP$_READBMAP                               ;
                          00000000G     00        04   FB 00559        CALLS   #4, LIB$SIGNAL                                  ;
                                        52   D4   00560 55$:           CLRL    J                                               ; 4916
                                        54   D4   00562 56$:           CLRL    K                                               ; 4919
                                50      14 AE42   D0   00564 57$:      MOVL    BUFFER[J], R0                                   ; 4925
                                        14 AE42   DF   00569           PUSHAL  BUFFER[J]                                       ; 4926
      51          9E                01        54   EF   00560        EXTZV   K, #1, @(SP)+, R1                               ;
                                21              57   E8   00572        BLBS    BITS_SET, 58$                                   ; 4922
                                                50   D5   00575        TSTL    R0                                              ; 4925
                                                71   13   00577        BEQL    60$                                             ;
                                68              51   E9   00579        BLBC    R1, 59$                                         ; 4926
                                57              01   D0   0057C        MOVL    #1, BITS_SET                                    ; 4929
                          50    53              0C   78   0057F        ASHL    #12, VBN, R0                                    ; 4930
                          51    52              05   78   00583        ASHL    #5, J, R1                                       ;
                                50              51   C0   00587        ADDL2   R1, R0                                          ;
                                50              54   C0   0058A        ADDL2   K, R0                                           ;
                                55      04      AB   3C   0058D        MOVZWL  4(VCB), FIRST_SET                               ;
                                55              50   C4   00591        MULL2   R0, FIRST_SET                                   ;
                                                4E   11   00594        BRB     59$                                             ; 4922
                             FFFFFFFF   8F              50   D1   00596 58$:     CMPL    R0, #-1                                 ; 4935
                                                4B   13   0059D        BEQL    60$                                             ;
                                42              51   E8   0059F        BLBS    R1, 59$                                         ; 4936
                                                57   D4   005A2        CLRL    BITS_SET                                        ; 4939
                          50    53              0C   78   005A4        ASHL    #12, VBN, R0                                    ; 4940
                          51    52              05   78   005A8        ASHL    #5, J, R1                                       ;
                                50              51   C0   005AC        ADDL2   R1, R0                                          ;
                                50              54   C0   005AF        ADDL2   K, R0                                           ;
                                51      04      AB   3C   005B2        MOVZWL  4(VCB), R1                                      ;
                    0C    AE          50              51   C5   005B6  MULL3   R1, R0, LAST_SET                                 ;
                                                    55   DD   005BB    PUSHL   FIRST_SET                                       ; 4941
                          7E      10    AE          55   C3   005BD    SUBL3   FIRST_SET, LAST_SET, -(SP)                      ;
                                ECFE    CF          02   FB   005C2    CALLS   #2, FREE_BLOCKS                                 ;
                                                    58   D6   005C7    INCL    EXTENT_COUNT                                    ; 4942
                             00000064   8F          58   D1   005C9    CMPL    EXTENT_COUNT, #100                              ; 4943
                                                    12   15   005D0    BLEQ    59$                                             ;
                                                20      AB   9F   005D2 PUSHAB  32(VCB)                                         ; 4944
                                                01      DD   005D5      PUSHL   #1                                              ;
                                         00000000G      8F   DD   005D7 PUSHL   #BACKUP$_DISKFRAG                               ;
                                00000000G       00          03   FB   005DD CALLS   #3, LIB$SIGNAL                             ;
      FF7A          54              01                    1F   F1   005E4 59$:     ACBL    #31, #1, K, 57$                 ; 4919
      FF6E          52              01      0000007F      8F   F1   005EA 60$:     ACBL    #127, #1, J, 56$                ; 4916
                    02              53                    56   F2   005F4 61$:     AOBLSS  COUNT, VBN, 62$                 ; 4902
                                                          03   11   005F8        BRB     63$                                     ;
                                                      FF1A   31   005FA 62$:     BRW     53$                                     ;
                                        16                  57   E9   005FD 63$:     BLBC    BITS_SET, 64$                   ; 4950
                                                            55   DD   00600        PUSHL   FIRST_SET                       ; 4951
                                        50      04          AB   3C   00602        MOVZWL  4(VCB), R0                      ;
                                        56                  50   C4   00606        MULL2   R0, R6                          ;
                          56            56                  0C   78   00609        ASHL    #12, R6, R6                     ;
                          7E            56                  55   C3   0060D        SUBL3   FIRST_SET, R6, -(SP)            ;
                                ECAF    CF                  02   FB   00611        CALLS   #2, FREE_BLOCKS                 ;
                                                        00AB   31   00616 64$:     BRW     69$                             ; 4797
                                        12      04          AC   E9   00619 65$:     BLBC    MODE, 66$                       ; 4959
```

```
                                    20   AB  9F 0061D              PUSHAB  32(VCB)                                    ; 4960
                                    01   DD  00620                PUSHL   #1                                          ;
                          00000000G      8F  DD  00622              PUSHL   #BACKUP$_ODS2SAVE                           ;
                00000000G   00           03  FB  00628              CALLS   #3, LIB$SIGNAL                              ;
                      04    AB           01  B0  0062F  66$:       MOVW    #1, 4(VCB)                                  ; 4961
          1A    AB    0214  CE           02  A1  00633              ADDW3   #2, HOME_BLOCK, 26(VCB)                     ; 4962
                        50    0214   CE   3C  0063A                MOVZWL  HOME_BLOCK, R0                              ; 4963
          1C    AB            50         0C  78  0063F              ASHL    #12, R0, 28(VCB)                            ;
                              7E         7C  00644                  CLRQ    -(SP)                                       ; 4970
                              7E         D4  00646                  CLRL    -(SP)                                       ;
                50    0222    CE         10  9C  00648              ROTL    #16, HOME_BLOCK+2, R0                       ;
                        51    0220   CE   3C  0064E                MOVZWL  HOME_BLOCK, R1                              ;
                                    6140   9F  00653                PUSHAB  (R1)[R0]                                    ;
                              7E    0200   8F  3C  00656            MOVZWL  #512, -(SP)                                 ;
                                    28     AE  9F  0065B            PUSHAB  HEADER                                      ;
                                           7E  7C  0065E            CLRQ    -(SP)                                       ;
                                    F8     AD  9F  00660            PUSHAB  IOSB                                        ;
                                           21  DD  00663            PUSHL   #33                                         ;
                                    30     AE  DD  00665            PUSHL   CHANNEL                                     ;
                                           7E  D4  00668            CLRL    -(SP)                                       ;
                00000000G   00           0C  FB  0066A              CALLS   #12, SYS$QIOW                               ;
                      04    AE           50  D0  00671              MOVL    R0, STATUS                                  ;
                            27     04    AE  E9  00675              BLBC    STATUS, 67$                                 ; 4971
                      04    AE     F8    AD  3C  00679              MOVZWL  IOSB, STATUS                                ;
                            1E     04    AE  E9  0067E              BLBC    STATUS, 67$                                 ; 4972
                                  F96E   CF  9F  00682              PUSHAB  INDEX_FILE_ID                               ; 4975
                                    18     AE  9F  00686            PUSHAB  HEADER                                      ;
                          E828      CF         02  FB  00689        CALLS   #2, VERIFY_HEADER                           ;
                      04    AE           50  D0  0068E              MOVL    R0, STATUS                                  ;
                            1F     04    AE  E8  00692              BLBS    STATUS, 68$                                 ; 4976
                      04    AE     0810   8F  3C  00696              MOVZWL  #2064, STATUS                               ;
                            15     04    AE  E8  0069C              BLBS    STATUS, 68$                                 ; 4978
                                    04     AE  DD  006A0  67$:       PUSHL   STATUS                                      ; 4980
                                    20     AB  9F  006A3              PUSHAB  32(VCB)                                     ;
                                    01     DD  006A6                  PUSHL   #1                                          ;
                          00000000G      8F  DD  006A8              PUSHL   #BACKUP$_NOINDEXF                           ;
                00000000G   00           04  FB  006AE              CALLS   #4, LIB$SIGNAL                              ;
                            7E         01  7D  006B5  68$:       MOVQ    #1, -(SP)                                   ; 4981
                                       5B  DD  006B8              PUSHL   VCB                                         ;
                                       01  DD  006BA              PUSHL   #1                                          ;
                                    24     AE  9F  006BC              PUSHAB  HEADER                                      ;
                          EDE6      CF         05  FB  006BF        CALLS   #5, CREATE_WINDOW                           ;
                                    02         6C  91  006C4  69$:       CMPB    (AP), #2                                    ; 4984
                                               07  1F  006C7              BLSSU   70$                                         ;
                50                  6E         31  C1  006C9              ADDL3   #49, MTL, R0                                ; 4985
                                    05         60  E8  006CD              BLBS    (R0), 71$                                   ;
                01                  5A     10  AE  F2  006D0  70$:       AOBLSS  16(SP), RVN, 72$                            ; 4677
                                               04  006D5  71$:       RET                                                 ; 4987
                                          FA6B 31  006D6  72$:       BRW     21$                                         ; 4677

; Routine Size:  1753 bytes,    Routine Base:  CODE + 1324
```

```
;  3463          4988  1 %SBTTL 'STA_DISMOUNT_OUTPUT - dismount output volume'
;  3464          4989  1 GLOBAL ROUTINE STA_DISMOUNT_OUTPUT (P_RVN, CONTINUE) : NOVALUE=
;  3465          4990  1
;  3466          4991  1 !++
;  3467          4992  1 !
;  3468          4993  1 ! FUNCTIONAL DESCRIPTION:
;  3469          4994  1 !       This routine is called at the completion of an image restore operation
;  3470          4995  1 !       and at the completion of each sequential disk output volume
;  3471          4996  1 !       to finish certain parts of the file structure:  index file bitmaps,
;  3472          4997  1 !       storage bitmaps, the quota file, and the VOLSET.SYS file.
;  3473          4998  1 !
;  3474          4999  1 ! INPUT PARAMETERS:
;  3475          5000  1 !       P_RVN (optional): if present, specifies RVN to dismount
;  3476          5001  1 !                         if absent, dismount entire set
;  3477          5002  1 !       CONTINUE (optional): 1 indicating save set volume will be continued
;  3478          5003  1 !                            0 indicating last volume in set
;  3479          5004  1 !
;  3480          5005  1 ! IMPLICIT INPUTS:
;  3481          5006  1 !       OUTPUT_MTL      - Pointer to MTL for output volume set.
;  3482          5007  1 !
;  3483          5008  1 ! OUTPUT PARAMETERS:
;  3484          5009  1 !       NONE
;  3485          5010  1 !
;  3486          5011  1 ! IMPLICIT OUTPUTS:
;  3487          5012  1 !       NONE
;  3488          5013  1 !
;  3489          5014  1 ! ROUTINE VALUE:
;  3490          5015  1 !       NONE
;  3491          5016  1 !
;  3492          5017  1 ! SIDE EFFECTS:
;  3493          5018  1 !       NONE
;  3494          5019  1 !
;  3495          5020  1 !--
;  3496          5021  1
;  3497          5022  2 BEGIN
;  3498          5023  2 LOCAL
;  3499          5024  2         RSA:            VECTOR[NAM$C_MAXRSS,BYTE],      ! Resultant string area
;  3500          5025  2         VCB:            REF BBLOCK,                     ! Pointer to VCB for volume
;  3501          5026  2         SAVE_WCB,                                       ! Saved window from MTL
;  3502          5027  2         BUFFER:         BITVECTOR[4096],                ! Block buffer
;  3503          5028  2         FIB:            BBLOCK[FIB$C_LENGTH],           ! FIB
;  3504          5029  2         FIB_DESC:       VECTOR[2],                      ! Descriptor for FIB
;  3505          5030  2         STATUS,                                         ! Status variable
;  3506          5031  2         IOSB:           VECTOR[4,WORD];                 ! I/O status block
;  3507          5032  2 BUILTIN
;  3508          5033  2         ACTUALCOUNT;
;  3509          5034  2
;  3510          5035  2
;  3511          5036  2 ! First scan the VCB's to make sure that all the volumes were
;  3512          5037  2 ! actually initialized. If we're dealing with a degenerate save set,
;  3513          5038  2 ! some may have been missed.
;  3514          5039  2 !
;  3515          5040  2 CURRENT_MTL = .OUTPUT_MTL;
;  3516          5041  2 INCR RVN
;  3517          5042  3 FROM (IF ACTUALCOUNT() EQL 0
;  3518          5043  3     THEN .CURRENT_MTL[MTL_RVN_BASE]
;  3519          5044  3     ELSE .P_RVN)
```

```
; 3520          5045  3 TO (IF ACTUALCOUNT() EQL 0
; 3521          5046  3       THEN .CURRENT_MTL[MTL_RVN_BASE]+.CURRENT_MTL[MTL_SETCOUNT]-1
; 3522          5047  3       ELSE .P_RVN)
; 3523          5048  2 DO
; 3524          5049  3       BEGIN
; 3525          5050  3       VCB = .CURRENT_MTL[MTL_VCB(.RVN-.CURRENT_MTL[MTL_RVN_BASE])];
; 3526          5051  3       IF NOT .VCB[VCB_INIT_DONE]
; 3527          5052  3       THEN SIGNAL (BACKUP$_NOVOLDATA, 1, VCB[VCB_DEVICE]);
; 3528          5053  2       END;
; 3529          5054  2
; 3530          5055  2 ! Save the accessed file, if any, and do other initialization.
; 3531          5056  2 !
; 3532          5057  2 SAVE_WCB = .CURRENT_MTL[MTL_WINDOW];
; 3533          5058  2 CURRENT_MTL[MTL_WINDOW] = 0;
; 3534          5059  2 RSA_DESC[1] = RSA;
; 3535          5060  2 FIB_DESC[0] = FIB$C_LENGTH;
; 3536          5061  2 FIB_DESC[1] = FIB;
; 3537          5062  2 CH$FILL (0, FIB$C_LENGTH, FIB);
; 3538          5063  2
; 3539          5064  2
; 3540          5065  2 ! Rebuild the quota file.
; 3541          5066  2 !
; 3542          5067  2 IF
; 3543          5068  2       .QUAL[QUAL_OF11] AND NOT .QUAL[QUAL_VOLU] AND
; 3544          5069  2       .DQF_QUOTA_FID[FID$W_NUM] NEQ 0
; 3545          5070  2 THEN
; 3546          5071  3       BEGIN
; 3547          5072  3       MAP
; 3548          5073  3           BUFFER:         BBLOCK[512];            ! Block buffer
; 3549          5074  3       LOCAL
; 3550          5075  3           FAT:            BBLOCK[FAT$C_LENGTH],   ! Quota file attributes
; 3551          5076  3           ATR_DESC:       BBLOCK[12];             ! ACP attributes list
; 3552          5077  3       GLOBAL REGISTER
; 3553          5078  3           DQF_BUFFER=             11,
; 3554          5079  3           DQF_RECORD=             10:     REF BBLOCK,
; 3555          5080  3           DQF_VBN=                9,
; 3556          5081  3           DQF_DEFAULT_PERM=       8,
; 3557          5082  3           DQF_DEFAULT_OVER=       7;
; 3558          5083  3
; 3559          5084  3
; 3560          5085  3       ! Access the quota file.
; 3561          5086  3       !
; 3562          5087  3       CURRENT_VCB = .CURRENT_MTL[MTL_VCB(0)];
; 3563          5088  3       RSA_DESC[0] = NAM$C_MAXRSS;
; 3564        P 5089  3       $FAO(
; 3565        P 5090  3           $DESCRIPTOR('!AS[000000]QUOTA.SYS;1'),
; 3566        P 5091  3           RSA_DESC,
; 3567        P 5092  3           RSA_DESC,
; 3568          5093  3           CURRENT_VCB[VCB_DEVICE]);
; 3569          5094  3       FIB[FIB$L_ACCTL] = FIB$M_WRITE OR FIB$M_NOWRITE;
; 3570          5095  3       FIB[FIB$W_FID_NUM] = .DQF_QUOTA_FID[FID$W_NUM];
; 3571          5096  3       FIB[FIB$W_FID_SEQ] = .DQF_QUOTA_FID[FID$W_SEQ];
; 3572          5097  3       FIB[FIB$W_FID_RVN] = .DQF_QUOTA_FID[FID$W_RVN];
; 3573          5098  3       ATR_DESC[0,0,16,0] = ATR$S_RECATTR;
; 3574          5099  3       ATR_DESC[2,0,16,0] = ATR$C_RECATTR;
; 3575          5100  3       ATR_DESC[4,0,32,0] = FAT;
; 3576          5101  3       ATR_DESC[8,0,32,0] = 0;
```

```
;  3577        P 5102  3      STATUS = $$QIOW(
;  3578        P 5103  3          FUNC=IO$_ACCESS OR IO$M_ACCESS,
;  3579        P 5104  3          CHAN=STA_OUT_CHAN,
;  3580        P 5105  3          IOSB=IOSB,
;  3581        P 5106  3          P1=FIB_DESC,
;  3582          5107  3          P5=ATR_DESC);
;  3583          5108  3      IF .STATUS THEN STATUS = .IOSB[0];
;  3584          5109  3      IF NOT .STATUS
;  3585          5110  3      THEN
;  3586          5111  4          BEGIN
;  3587          5112  4          SIGNAL(BACKUP$_OPENOUT + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;  3588          5113  4          RETURN;
;  3589          5114  4          END;
;  3590          5115  3
;  3591          5116  3
;  3592          5117  3      ! Scan the quota file to record the quotas in the quota table.
;  3593          5118  3      !
;  3594          5119  3      DQF_DEFAULT_PERM = 0;
;  3595          5120  3      DQF_DEFAULT_OVER = 0;
;  3596          5121  3      INCR VBN FROM 1 TO ROT(.FAT[FAT$L_EFBLK], 16) - 1 DO
;  3597          5122  4          BEGIN
;  3598          5123  4
;  3599          5124  4          ! Read a block of the quota file.
;  3600          5125  4          !
;  3601        P 5126  4          STATUS = $$QIOW(
;  3602        P 5127  4              FUNC=IO$_READVBLK,
;  3603        P 5128  4              CHAN=STA_OUT_CHAN,
;  3604        P 5129  4              IOSB=IOSB,
;  3605        P 5130  4              P1=BUFFER,
;  3606        P 5131  4              P2=512,
;  3607          5132  4              P3=.VBN);
;  3608          5133  4          IF .STATUS THEN STATUS = .IOSB[0];
;  3609          5134  4          IF NOT .STATUS
;  3610          5135  4          THEN
;  3611          5136  4              SIGNAL(BACKUP$_READERR + STS$K_ERROR, 1, RSA_DESC, .STATUS)
;  3612          5137  4          ELSE
;  3613          5138  4              INCRA P FROM BUFFER TO BUFFER+512-DQF$C_LENGTH BY DQF$C_LENGTH DO
;  3614          5139  5                  BEGIN
;  3615          5140  5                  MAP
;  3616          5141  5                      P:          REF BBLOCK;     ! Pointer to quota file entry
;  3617          5142  5                  LOCAL
;  3618          5143  5                      Q:          REF BBLOCK;     ! Pointer to quota table entry
;  3619          5144  5
;  3620          5145  5
;  3621          5146  5                  ! Scan the quota entries in the block to extract the quotas.
;  3622          5147  5                  !
;  3623          5148  5                  IF .P[DQF$V_ACTIVE]
;  3624          5149  5                  THEN
;  3625          5150  6                      BEGIN
;  3626          5151  6                      IF .P[DQF$L_UIC] EQL 0
;  3627          5152  6                      THEN
;  3628          5153  7                          BEGIN
;  3629          5154  7                          DQF_DEFAULT_PERM = .P[DQF$L_PERMQUOTA];
;  3630          5155  7                          DQF_DEFAULT_OVER = .P[DQF$L_OVERDRAFT];
;  3631          5156  6                          END;
;  3632          5157  6                      Q = DQF_FIND_UIC(.P[DQF$L_UIC]);
;  3633          5158  6                      Q[DQF_PERMQUOTA] = .P[DQF$L_PERMQUOTA];
```

```
;  3634          5159  6                    Q[DQF_OVERDRAFT] = .P[DQF$L_OVERDRAFT];
;  3635          5160  5                    END;
;  3636          5161  4                END;
;  3637          5162  4            END;
;  3638          5163  4
;  3639          5164  4
;  3640          5165  4            ! Ensure that the allocated size of the quota file is sufficient for the
;  3641          5166  4            ! records to be written into it.
;  3642          5167  4            !
;  3643          5168  4            IF ROT(.FAT[FAT$L_HIBLK], 16) LSSU .DQF_COUNT / (512/DQF$C_LENGTH)
;  3644          5169  3            THEN
;  3645          5170  4                BEGIN
;  3646          5171  4                SIGNAL(BACKUP$_QUOTAFILE);
;  3647        P 5172  4                S$QIOW(
;  3648        P 5173  4                    FUNC=IO$_DEACCESS,
;  3649          5174  4                    CHAN=STA_OUT_CHAN);
;  3650          5175  4                RETURN;
;  3651          5176  4                END;
;  3652          5177  4
;  3653          5178  4
;  3654          5179  4            ! Rewrite the quota file.
;  3655          5180  4            !
;  3656          5181  4            CH$FILL(0, 512, BUFFER);
;  3657          5182  4            DQF_BUFFER = DQF_RECORD = BUFFER;
;  3658          5183  4            DQF_VBN = 1;
;  3659          5184  4            IF .DQF_ROOT NEQ 0 THEN DQF_WRITE_ENTRY(.DQF_ROOT);
;  3660          5185  4
;  3661          5186  4
;  3662          5187  4            ! Flush out the last block.
;  3663          5188  4            !
;  3664          5189  4            IF .DQF_RECORD NEQ .DQF_BUFFER
;  3665          5190  3            THEN
;  3666          5191  4                BEGIN
;  3667        P 5192  4                STATUS = S$QIOW(
;  3668        P 5193  4                    FUNC=IO$_WRITEVBLK,
;  3669        P 5194  4                    CHAN=STA_OUT_CHAN,
;  3670        P 5195  4                    IOSB=IOSB,
;  3671        P 5196  4                    P1=.DQF_BUFFER,
;  3672        P 5197  4                    P2=512,
;  3673          5198  4                    P3=.DQF_VBN);
;  3674          5199  4                DQF_VBN = .DQF_VBN + 1;
;  3675          5200  4                IF .STATUS THEN STATUS = .IOSB[0];
;  3676          5201  4                IF NOT .STATUS
;  3677          5202  4                THEN
;  3678          5203  4                    SIGNAL(BACKUP$_WRITEERR + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;  3679          5204  4                END;
;  3680          5205  4
;  3681          5206  4
;  3682          5207  4            ! Deaccess the quota file and rewrite the end of file pointer.
;  3683          5208  4            !
;  3684          5209  4            FAT[FAT$L_EFBLK] = ROT(.DQF_VBN, 16);
;  3685        P 5210  4            STATUS = S$QIOW(
;  3686        P 5211  4                FUNC=IO$_DEACCESS,
;  3687        P 5212  4                CHAN=STA_OUT_CHAN,
;  3688        P 5213  4                IOSB=IOSB,
;  3689          5214  4                P5=ATR_DESC);
;  3690          5215  3            IF .STATUS THEN STATUS = .IOSB[0];
```

```
;3691        5216  3      IF NOT .STATUS
;3692        5217  3      THEN
;3693        5218  3          SIGNAL(BACKUP$_CLOSEOUT + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;3694        5219  2      END;
;3695        5220  2
;3696        5221  2
;3697        5222  2  ! Finish the VOLSET.SYS file.  It may be incorrect if any volume labels were
;3698        5223  2  ! changed by /NOINITIALIZE.
;3699        5224  2  !
;3700        5225  2  IF
;3701        5226  2      .CURRENT_MTL[MTL_SETCOUNT] GTRU 1 AND
;3702        5227  2      ACTUALCOUNT() EQL 0
;3703        5228  2  THEN
;3704        5229  3      BEGIN
;3705        5230  3      LOCAL
;3706        5231  3          VBN,
;3707        5232  3          P;
;3708        5233  3      MAP
;3709        5234  3          BUFFER:         VECTOR[,BYTE];
;3710        5235  3
;3711        5236  3
;3712        5237  3      ! Access VOLSET.SYS file on RVN 1.
;3713        5238  3      !
;3714        5239  3      CURRENT_VCB = .CURRENT_MTL[MTL_VCB(0)];
;3715        5240  3      RSA_DESC[0] = NAM$C_MAXRSS;
;3716      P 5241  3      $FAO(
;3717      P 5242  3          $DESCRIPTOR('!AS[000000]VOLSET.SYS;1'),
;3718      P 5243  3          RSA_DESC,
;3719      P 5244  3          RSA_DESC,
;3720        5245  3          CURRENT_VCB[VCB_DEVICE]);
;3721        5246  3      FIB[FIB$L_ACCTL] = FIB$M_WRITE OR FIB$M_NOWRITE;
;3722        5247  3      FIB[FIB$W_FID_NUM] = FID$C_VOLSET;
;3723        5248  3      FIB[FIB$W_FID_SEQ] = FID$C_VOLSET;
;3724        5249  3      FIB[FIB$W_FID_RVN] = 1;
;3725      P 5250  3      STATUS = $$QIOW(
;3726      P 5251  3          FUNC=IO$_ACCESS OR IO$M_ACCESS,
;3727      P 5252  3          CHAN=STA_OUT_CHAN,
;3728      P 5253  3          IOSB=IOSB,
;3729        5254  3          P1=FIB_DESC);
;3730        5255  3      IF .STATUS THEN STATUS = .IOSB[0];
;3731        5256  3      IF NOT .STATUS
;3732        5257  3      THEN
;3733        5258  3          SIGNAL(BACKUP$_OPENOUT + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;3734        5259  3
;3735        5260  3
;3736        5261  3      ! Initialize volume set name.
;3737        5262  3      !
;3738        5263  3      CH$COPY(VSL$S_NAME, COM_O_STRUCNAME, 0, 512, BUFFER);
;3739        5264  3      P = VSL$C_LENGTH;
;3740        5265  3      VBN = 1;
;3741        5266  3
;3742        5267  3
;3743        5268  3      INCR RVN FROM 1 TO .CURRENT_MTL[MTL_SETCOUNT] DO
;3744        5269  4          BEGIN
;3745        5270  4
;3746        5271  4          ! If block is full, write it and reinitialize.
;3747        5272  4          !
```

```
;  3748          5273  4          IF .P GEQU 512
;  3749          5274  4          THEN
;  3750          5275  5              BEGIN
;  3751        P 5276  5              STATUS = $$QIOW(
;  3752        P 5277  5                  FUNC=IO$_WRITEVBLK,
;  3753        P 5278  5                  CHAN=STA_OUT_CHAN,
;  3754        P 5279  5                  IOSB=IOSB,
;  3755        P 5280  5                  P1=BUFFER,
;  3756        P 5281  5                  P2=512,
;  3757          5282  5                  P3=.VBN);
;  3758          5283  5              IF .STATUS THEN STATUS = .IOSB[0];
;  3759          5284  5              IF NOT .STATUS
;  3760          5285  5              THEN
;  3761          5286  5                  SIGNAL(BACKUP$_WRITEERR + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;  3762          5287  5
;  3763          5288  5
;  3764          5289  5              ! Reinitialize buffer, pointer, counts.
;  3765          5290  5              !
;  3766          5291  5              CH$FILL(0, 512, BUFFER);
;  3767          5292  5              P = 0;
;  3768          5293  5              VBN = .VBN + 1;
;  3769          5294  4              END;
;  3770          5295  4
;  3771          5296  4
;  3772          5297  4          ! Initialize VSL entry.
;  3773          5298  4          !
;  3774          5299  4          CH$MOVE(
;  3775          5300  4              VSL$S_NAME,
;  3776          5301  4              BBLOCK[.CURRENT_MTL[MTL_VCB(.RVN-.CURRENT_MTL[MTL_RVN_BASE])], VCB_VOLNAME],
;  3777          5302  4              BBLOCK[BUFFER[.P], VSL$T_NAME]);
;  3778          5303  4
;  3779          5304  4
;  3780          5305  4              P = .P + VSL$C_LENGTH;
;  3781          5306  3              END;
;  3782          5307  3
;  3783          5308  3
;  3784          5309  3          ! Write last block.
;  3785          5310  3          !
;  3786        P 5311  3          STATUS = $$QIOW(
;  3787        P 5312  3              FUNC=IO$_WRITEVBLK,
;  3788        P 5313  3              CHAN=STA_OUT_CHAN,
;  3789        P 5314  3              IOSB=IOSB,
;  3790        P 5315  3              P1=BUFFER,
;  3791        P 5316  3              P2=512,
;  3792          5317  3              P3=.VBN);
;  3793          5318  3          IF .STATUS THEN STATUS = .IOSB[0];
;  3794          5319  3          IF NOT .STATUS
;  3795          5320  3          THEN
;  3796          5321  3              SIGNAL(BACKUP$_WRITEERR + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;  3797          5322  3
;  3798          5323  3
;  3799          5324  3          ! Deaccess the file.
;  3800          5325  3          !
;  3801        P 5326  3          $$QIOW(
;  3802        P 5327  3              FUNC=IO$_DEACCESS,
;  3803          5328  3              CHAN=STA_OUT_CHAN);
;  3804          5329  2          END;
```

```
;  3805          5330  2
;  3806          5331  2
;  3807          5332  2 ! If this is RVN 1 of a save set, it may be an externally initialized
;  3808          5333  2 ! volume.  If it is being continued, we want to mark it as RVN 1 of a
;  3809          5334  2 ! loosely coupled volume set.  Loop through the home blocks to do this.
;  3810          5335  2 !
;  3811          5336  2 CURRENT_VCB = .CURRENT_MTL[MTL_VCB(0)];
;  3812          5337  2 IF ACTUALCOUNT () GEQ 2
;  3813          5338  2 AND .P_RVN EQL 1
;  3814          5339  2 AND .CONTINUE
;  3815          5340  2 AND .CURRENT_VCB[VCB_NOTVOLSET]
;  3816          5341  2 THEN
;  3817          5342  3     BEGIN
;  3818          5343  3     MAP
;  3819          5344  3         BUFFER:         BBLOCK;                 ! Block buffer
;  3820          5345  3
;  3821          5346  3
;  3822          5347  3     ! Access the index file on RVN 1.
;  3823          5348  3     !
;  3824          5349  3     CURRENT_MTL[MTL_WINDOW] = .CURRENT_VCB[VCB_INDEXF];
;  3825          5350  3     RSA_DESC[0] = NAM$C_MAXRSS;
;  3826        P 5351  3     $FAO(
;  3827        P 5352  3         $DESCRIPTOR('!AS[000000]INDEXF.SYS;1'),
;  3828        P 5353  3         RSA_DESC,
;  3829        P 5354  3         RSA_DESC,
;  3830          5355  3         CURRENT_VCB[VCB_DEVICE]);
;  3831          5356  3     CURRENT_WCB = .CURRENT_MTL[MTL_WINDOW];
;  3832          5357  3
;  3833          5358  3
;  3834          5359  3 ! Loop, reading home blocks from the index file.
;  3835          5360  3 !
;  3836          5361  3
;  3837          5362  3     INCR VBN FROM 2 TO .CURRENT_VCB[VCB_CLUSTER]*3
;  3838          5363  3         DO
;  3839          5364  4         BEGIN
;  3840          5365  4         STATUS = R_W_VIRTUAL(
;  3841          5366  4             0,
;  3842          5367  4             0,
;  3843          5368  4             IO$_READVBLK,
;  3844          5369  4             IOSB,
;  3845          5370  4             0,
;  3846          5371  4             0,
;  3847          5372  4             BUFFER,
;  3848          5373  4             512,
;  3849          5374  4             .VBN);
;  3850          5375  4         $WAITFR(EFN=0);
;  3851          5376  4         IF .STATUS THEN STATUS = .IOSB[0];
;  3852          5377  4         IF NOT .STATUS
;  3853          5378  4         THEN
;  3854          5379  4             SIGNAL(BACKUP$_READERR + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;  3855          5380  4
;  3856          5381  4
;  3857          5382  4         IF NOT CHECKSUM2 (BUFFER, $BYTEOFFSET (HM2$W_CHECKSUM1))
;  3858          5383  4         OR NOT CHECKSUM2 (BUFFER, $BYTEOFFSET (HM2$W_CHECKSUM2))
;  3859          5384  4         THEN SIGNAL(BACKUP$_INVHOMBLK, 1, RSA_DESC);
;  3860          5385  4
;  3861          5386  4
```

```
; 3862          5387  4 ! Write RVN and volume set name into home block and re-checksum.
; 3863          5388  4 !
; 3864          5389  4
; 3865          5390  4         BUFFER[HM2$W_RVN] = 1;
; 3866          5391  4         CH$MOVE (HM2$S_STRUCNAME, CURRENT_MTL[MTL_STRUCNAME], BUFFER[HM2$T_STRUCNAME]);
; 3867          5392  4         CHECKSUM2 (BUFFER, $BYTEOFFSET (HM2$W_CHECKSUM1));
; 3868          5393  4         CHECKSUM2 (BUFFER, $BYTEOFFSET (HM2$W_CHECKSUM2));
; 3869          5394  4
; 3870          5395  4 ! Write back the updated home block.
; 3871          5396  4 !
; 3872          5397  4
; 3873          5398  4         STATUS = R_W_VIRTUAL(
; 3874          5399  4             0,
; 3875          5400  4             0,
; 3876          5401  4             IO$_WRITEVBLK,
; 3877          5402  4             IOSB,
; 3878          5403  4             0,
; 3879          5404  4             0,
; 3880          5405  4             BUFFER,
; 3881          5406  4             512,
; 3882          5407  4             .VBN);
; 3883          5408  4         $WAITFR(EFN=0);
; 3884          5409  4         IF .STATUS THEN STATUS = .IOSB[0];
; 3885          5410  4         IF NOT .STATUS
; 3886          5411  4         THEN
; 3887          5412  4             SIGNAL(BACKUP$_WRITEERR + STS$K_ERROR, 1, RSA_DESC, .STATUS);
; 3888          5413  3         END;
; 3889          5414  3
; 3890          5415  3
; 3891          5416  3     CURRENT_MTL[MTL_WINDOW] = 0;
; 3892          5417  2     END;
; 3893          5418  2
; 3894          5419  2
; 3895          5420  2 ! Do each volume in the volume set.
; 3896          5421  2 !
; 3897          5422  2 INCR RVN
; 3898          5423  2 FROM (IF ACTUALCOUNT() EQL 0
; 3899          5424  2     THEN .CURRENT_MTL[MTL_RVN_BASE]
; 3900          5425  2     ELSE .P_RVN)
; 3901          5426  2 TO (IF ACTUALCOUNT() EQL 0
; 3902          5427  2     THEN .CURRENT_MTL[MTL_RVN_BASE]+.CURRENT_MTL[MTL_SETCOUNT]-1
; 3903          5428  2     ELSE .P_RVN)
; 3904          5429  2 DO
; 3905          5430  2     BEGIN
; 3906          5431  2     LOCAL
; 3907          5432  2         VCB:            REF BBLOCK,             ! Pointer to VCB
; 3908          5433  2         CHANNEL,                                ! Channel assigned to volume
; 3909          5434  2         OFFSET,                                 ! Offset to bitmap block
; 3910          5435  2         ACB:            REF BBLOCK;             ! Pointer to ACB
; 3911          5436  2
; 3912          5437  2
; 3913          5438  2     CURRENT_VCB = VCB = .CURRENT_MTL[MTL_VCB(.RVN-.CURRENT_MTL[MTL_RVN_BASE])];
; 3914          5439  2     RSA_DESC[0] = NAM$C_MAXRSS;
; 3915        P 5440  2     $FAO(
; 3916        P 5441  2         $DESCRIPTOR('!AS[000000]INDEXF.SYS;1'),
; 3917        P 5442  2         RSA_DESC,
; 3918        P 5443  3         RSA_DESC,
```

```
;3919          5444  3          VCB[VCB_DEVICE]);
;3920          5445  3      CHANNEL = SWITCH_VOLUME(.RVN);
;3921          5446  3
;3922          5447  3
;3923          5448  3      ! Write the index file bitmap.
;3924          5449  3      !
;3925        P 5450  3      STATUS = $QIOW(
;3926        P 5451  3          FUNC=IO$_WRITELBLK,
;3927        P 5452  3          CHAN=.CHANNEL,
;3928        P 5453  3          IOSB=IOSB,
;3929        P 5454  3          P1=.VCB[VCB_IMAP],
;3930        P 5455  3          P2=.VCB[VCB_MAXFILIDX] / 8,
;3931          5456  3          P3=.VCB[VCB_IMAP_LBN]);
;3932          5457  3      IF .STATUS THEN STATUS = .IOSB[0];
;3933          5458  3      IF NOT .STATUS
;3934          5459  3      THEN
;3935          5460  3          SIGNAL(BACKUP$_WRITEERR + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;3936          5461  3
;3937          5462  3
;3938          5463  3      ! Initialize for writing storage bitmap.
;3939          5464  3      !
;3940          5465  3      OFFSET = -1;
;3941          5466  3      ACB = .VCB[VCB_ACB_FLINK];
;3942          5467  3      RSA_DESC[0] = NAM$C_MAXRSS;
;3943        P 5468  3      $FAO(
;3944        P 5469  3          $DESCRIPTOR('!AS[000000]BITMAP.SYS;1'),
;3945        P 5470  3          RSA_DESC,
;3946        P 5471  3          RSA_DESC,
;3947          5472  3          VCB[VCB_DEVICE]);
;3948          5473  3
;3949          5474  3
;3950          5475  3      ! Loop for each ACB in the queue.
;3951          5476  3      !
;3952          5477  3      UNTIL REMQUE(.VCB[VCB_ACB_FLINK], ACB) DO
;3953          5478  4          BEGIN
;3954          5479  4
;3955          5480  4          ! Loop for each affected bit in the bitmap.  During the loop, N is the
;3956          5481  4          ! offset within the storage bitmap of the affected bit.
;3957          5482  4          !
;3958          5483  4          INCRU N
;3959          5484  4          FROM .ACB[ACB_LBN] / .VCB[VCB_CLUSTER]
;3960          5485  4          TO .ACB[ACB_LBN] / .VCB[VCB_CLUSTER] + .ACB[ACB_COUNT] / .VCB[VCB_CLUSTER] - 1
;3961          5486  4          DO
;3962          5487  5              BEGIN
;3963          5488  5
;3964          5489  5              ! If the current bit is not within the current bitmap block,
;3965          5490  5              ! write out the completed block, if any, initialize a new one,
;3966          5491  5              ! and pass zero blocks until we reach the right one.
;3967          5492  5              !
;3968          5493  5              UNTIL .N<12,20> LEQ .OFFSET
;3969          5494  5              DO
;3970          5495  6                  BEGIN
;3971          5496  6                  IF .OFFSET NEQ -1
;3972          5497  6                  THEN
;3973          5498  7                      BEGIN
;3974        P 5499  7                      STATUS = $QIOW(
;3975        P 5500  7                          FUNC=IO$_WRITELBLK,
```

```
;  3976        P 5501  7                           CHAN=.CHANNEL,
;  3977        P 5502  7                           IOSB=IOSB,
;  3978        P 5503  7                           P1=BUFFER,
;  3979        P 5504  7                           P2=512,
;  3980          5505  7                           P3=.VCB[VCB_BITMAP_LBN] + .OFFSET);
;  3981          5506  7                       IF .STATUS THEN STATUS = .IOSB[0];
;  3982          5507  7                       IF NOT .STATUS
;  3983          5508  7                       THEN
;  3984          5509  7                           SIGNAL(BACKUP$_WRITEERR + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;  3985          5510  6                       END;
;  3986          5511  6                   CH$FILL(0, 512, BUFFER);
;  3987          5512  6                   OFFSET = .OFFSET + 1;
;  3988          5513  5                   END;
;  3989          5514  5
;  3990          5515  5
;  3991          5516  5               ! Set the appropriate bit to record the free cluster.
;  3992          5517  5               !
;  3993          5518  5               BUFFER[.N<0,12>] = TRUE;
;  3994          5519  4               END;
;  3995          5520  4
;  3996          5521  4
;  3997          5522  4           ! Free the ACB.
;  3998          5523  4           !
;  3999          5524  4           FREE_VM(ACB_S_ENTRY, .ACB);
;  4000          5525  3           END;
;  4001          5526  3
;  4002          5527  3
;  4003          5528  3       ! Flush the last block, if any.
;  4004          5529  3       !
;  4005          5530  3       UNTIL .OFFSET GEQ .VCB[VCB_BITMAP_SIZE]
;  4006          5531  3       DO
;  4007          5532  4           BEGIN
;  4008          5533  4           IF .OFFSET NEQ -1
;  4009          5534  4           THEN
;  4010          5535  5               BEGIN
;  4011        P 5536  5               STATUS = $QIOW(
;  4012        P 5537  5                   FUNC=IO$_WRITELBLK,
;  4013        P 5538  5                   CHAN=.CHANNEL,
;  4014        P 5539  5                   IOSB=IOSB,
;  4015        P 5540  5                   P1=BUFFER,
;  4016        P 5541  5                   P2=512,
;  4017          5542  5                   P3=.VCB[VCB_BITMAP_LBN] + .OFFSET);
;  4018          5543  5               IF .STATUS THEN STATUS = .IOSB[0];
;  4019          5544  5               IF NOT .STATUS
;  4020          5545  5               THEN
;  4021          5546  5                   SIGNAL(BACKUP$_WRITEERR + STS$K_ERROR, 1, RSA_DESC, .STATUS);
;  4022          5547  4               END;
;  4023          5548  4           CH$FILL (0, 512, BUFFER);
;  4024          5549  4           OFFSET = .OFFSET + 1;
;  4025          5550  3           END;
;  4026          5551  3
;  4027          5552  3
;  4028          5553  3       ! Free the index file bitmap buffers.
;  4029          5554  3       !
;  4030          5555  3       FREE_VM(.VCB[VCB_MAXFILIDX]/8, .VCB[VCB_IMAP]);
;  4031          5556  3       VCB[VCB_IMAP] = 0;
;  4032          5557  2       END;
```

```
;  4033          5558  2
;  4034          5559  2
;  4035          5560  2 ! Restore the accessed file, if any.
;  4036          5561  2 !
;  4037          5562  2 CURRENT_MTL[MTL_WINDOW] = .SAVE_WCB;
;  4038          5563  1 END;


54 4F 55 51 5D 30 30 30 30 30 30 5B 53 41 21  019FD P.AAJ:  .ASCII  \!AS[000000]QUOTA.SYS;1\          ;
                                  31 3B 53 59 53 2E 41  01A0C                                        ;
                                                01A13          .BLKB   1                           ;
                                 00000016       01A14 P.AAI:  .LONG   22                          ;
                                 00000000'      01A18          .ADDRESS P.AAJ                      ;
53 4C 4F 56 5D 30 30 30 30 30 30 5B 53 41 21  01A1C P.AAL:  .ASCII  \!AS[000000]VOLSET.SYS;1\       ;
                                  31 3B 53 59 53 2E 54 45  01A2B                                     ;
                                                01A33          BLKB   1                            ;
                                 00000017       01A34 P.AAK:  .LONG   23                          ;
                                 00000000'      01A38          .ADDRESS P.AAL                      ;
45 44 4E 49 5D 30 30 30 30 30 30 5B 53 41 21  01A3C P.AAN:  .ASCII  \!AS[000000]INDEXF.SYS;1\       ;
                                  31 3B 53 59 53 2E 46 58  01A4B                                     ;
                                                01A53          .BLKB   1                           ;
                                 00000017       01A54 P.AAM:  .LONG   23                          ;
                                 00000000'      01A58          .ADDRESS P.AAN                      ;
45 44 4E 49 5D 30 30 30 30 30 30 5B 53 41 21  01A5C P.AAP:  .ASCII  \!AS[000000]INDEXF.SYS;1\       ;
                                  31 3B 53 59 53 2E 46 58  01A6B                                     ;
                                                01A73          .BLKB   1                           ;
                                 00000017       01A74 P.AAO:  .LONG   23                          ;
                                 00000000'      01A78          .ADDRESS P.AAP                      ;
4D 54 49 42 5D 30 30 30 30 30 30 5B 53 41 21  01A7C P.AAR:  .ASCII  \!AS[000000]BITMAP.SYS;1\       ;
                                  31 3B 53 59 53 2E 50 41  01A8B                                     ;
                                                01A93          .BLKB   1                           ;
                                 00000017       01A94 P.AAQ:  .LONG   23                          ;
                                 00000000'      01A98          .ADDRESS P.AAR                      ;


                              0FFC 00000                .ENTRY  STA_DISMOUNT_OUTPUT, Save R2,R3,R4,R5,R6,-  ; 4989
                                                                R7,R8,R9,R10,R11                    ;
                    5E     FC74   CE  9E 00002         MOVAB   -908(SP), SP                        ;
00000000'           EF 00000000'  EF  D0 00007         MOVL    OUTPUT_MTL, CURRENT_MTL             ; 5040
                                  6C  95 00012         TSTB    (AP)                                ; 5042
                                  0D  12 00014         BNEQ    1$                                  ;
                    50 00000000'  EF  D0 00016         MOVL    CURRENT_MTL, R0                     ; 5043
                    52       30   A0  9A 0001D         MOVZBL  48(R0), R2                          ;
                                  04  11 00021         BRB     2$                                  ;
                    52       04   AC  D0 00023 1$:     MOVL    P_RVN, R2                           ; 5044
                                  6C  95 00027 2$:     TSTB    (AP)                                ; 5045
                                  17  12 00029         BNEQ    3$                                  ;
                    50 00000000'  EF  D0 0002B         MOVL    CURRENT_MTL, R0                     ; 5046
                    51       30   A0  9A 00032         MOVZBL  48(R0), R1                          ;
                    50       1F   A0  9A 00036         MOVZBL  31(R0), R0                          ;
                    50            51  C0 0003A         ADDL2   R1, R0                              ;
                    54            70  9E 0003D         MOVAB   -(R0), R4                           ;
                                  04  11 00040         BRB     4$                                  ;
                    54       04   AC  D0 00042 3$:     MOVL    P_RVN, R4                           ; 5047
                                  52  D7 00046 4$:     DECL    RVN                                 ; 5041
```

```
                                                2B  11 00048          BRB     6$
                                    51 00000000'  EF  D0 0004A 5$:     MOVL    CURRENT_MTL, R1                         ;5050
                                    50       30   A1  9A 00051         MOVZBL  48(R1), R0
                                50  52            50  C3 00055         SUBL3   R0, RVN, R0
                                    53       34 A140 D0 00059         MOVL    52(R1)[R0], VCB
                                12       07   A3      02 E0 0005E         BBS     #2, 7(VCB), 6$                  ;5051
                                              20   A3  9F 00063         PUSHAB  32(VCB)                         ;5052
                                              01   DD  00066         PUSHL   #1
                                    00000000G  8F   DD  00068         PUSHL   #BACKUP$_NOVOLDATA
                                00000000G  00       03  FB 0006E         CALLS   #3, LIB$SIGNAL
                                D1  52              54  F3 00075 6$:     AOBLEQ  R4, RVN, 5$                     ;5041
                                    57 00000000'  EF  D0 00079         MOVL    CURRENT_MTL, R7                 ;5057
                                       0C   AE      08  A7 D0 00080         MOVL    8(R7), SAVE_WCB
                                              08   A7  D4 00085         CLRL    8(R7)                           ;5058
                                00000000'  EF  FF00   CD  9E 00088         MOVAB   RSA, RSA_DESC+4                 ;5059
                                       44   AE    40   8F  9A 00091         MOVZBL  #64, FIB_DESC                   ;5060
                                       48   AE    4C   AE  9E 00096         MOVAB   FIB, FIB_DESC+4                 ;5061
                            0040  8F   00       6E       00  2C 0009B         MOVC5   #0, (SP), #0, #64, FIB          ;5062
                                              4C   AE      000A2
                                03 00000000'  EF      06  E0 000A4         BBS     #6, QUAL+15, 8$                 ;5068
                                                  022D  31 000AC 7$:     BRW     24$
                                    F6 00000000'  EF  E8 000AF 8$:     BLBS    QUAL+14, 7$
                                       00000000'  EF  B5 00086         TSTW    DQF_QUOTA_FID                   ;5069
                                                  EE  13 000BC         BEQL    7$
                                00000000'  EF       34   A7  D0 000BE         MOVL    52(R7), CURRENT_VCB             ;5087
                                00000000'  EF       FF   8F  9A 000C6         MOVZBL  #255, RSA_DESC                  ;5088
                                7E 00000000'  EF            20  C1 000CE         ADDL3   #32, CURRENT_VCB, -(SP)         ;5093
                                       00000000'  EF  9F 000D6         PUSHAB  RSA_DESC
                                       00000000'  EF  9F 000DC         PUSHAB  RSA_DESC
                                           FE92   CF  9F 000E2         PUSHAB  P.AAI
                                00000000G  00          04  FB 000E6         CALLS   #4, SYS$FAO
                                      4C   AE     0101   8F  3C 000ED         MOVZWL  #257, FIB                       ;5094
                                      50   AE 00000000'  EF  D0 000F3         MOVL    DQF_QUOTA_FID, FIB+4            ;5095
                                      54   AE 00000000'  EF  B0 000FB         MOVW    DQF_QUOTA_FID+4, FIB+8          ;5097
                                      10   AE 00040020   8F  D0 00103         MOVL    #262176, ATR_DESC               ;5098
                                      14   AE       1C   AE  9E 0010B         MOVAB   FAT, ATR_DESC+4                 ;5100
                                                18   AE  D4 00110         CLRL    ATR_DESC+8                      ;5101
                                                7E   D4  00113         CLRL    -(SP)                           ;5107
                                                14   AE  9F 00115         PUSHAB  ATR_DESC
                                                7E   7C  00118         CLRQ    -(SP)
                                                7E   D4  0011A         CLRL    -(SP)
                                                58   AE  9F 0011C         PUSHAB  FIB_DESC
                                                7E   7C  0011F         CLRQ    -(SP)
                                                5C   AE  9F 00121         PUSHAB  IOSB
                                           7E       72   8F  9A 00124         MOVZBL  #114, -(SP)
                                              0002FFFF   8F  DD 00128         PUSHL   #196607
                                                7E   D4  0012E         CLRL    -(SP)
                                00000000G  00          0C  FB 00130         CALLS   #12, STA_QIOW
                                           56          50  D0 00137         MOVL    R0, STATUS
                                           07          56  E9 0013A         BLBC    STATUS, 9$                      ;5108
                                           56       3C   AE  3C 0013D         MOVZWL  IOSB, STATUS
                                           18          56  E8 00141         BLBS    STATUS, 10$                     ;5109
                                                       56  DD 00144 9$:     PUSHL   STATUS                          ;5112
                                              00000000'  EF  9F 00146         PUSHAB  RSA_DESC
                                                         01  DD 0014C         PUSHL   #1
                                              00000000G  8F  DD 0014E         PUSHL   #BACKUP$_OPENOUT+2
                                00000000G  00             04  FB 00154         CALLS   #4, LIB$SIGNAL
```

```
                                                 04 0015B          RET                                                 ;5111
                                              57 7C 0015C 10$:     CLRQ    DQF_DEFAULT_OVER                            ;5120
                    55      24    AE          10 9C 0015E          ROTL    #16, FAT+8, R5                              ;5121
                                              53 D4 00163          CLRL    VBN
                                              7C 11 00165          BRB     18$
                                              7E 7C 00167 11$:     CLRQ    -(SP)                                       ;5132
                                              7E D4 00169          CLRL    -(SP)
                                              53 DD 0016B          PUSHL   VBN
                                 7E     0200  8F 3C 0016D          MOVZWL  #512, -(SP)
                                       00A0  CE 9F 00172          PUSHAB  BUFFER
                                              7E 7C 00176          CLRQ    -(SP)
                                        5C    AE 9F 00178          PUSHAB  IOSB
                                              31 DD 0017B          PUSHL   #49
                                  0002FFFF    8F DD 0017D          PUSHL   #196607
                                              7E D4 00183          CLRL    -(SP)
                00000000G         00          0C FB 00185          CALLS   #12, STA_QIOW
                                  56          50 D0 0018C          MOVL    R0, STATUS                                  ;5133
                                  07          56 E9 0018F          BLBC    STATUS, 12$                                 ;5134
                                  56     3C   AE 3C 00192          MOVZWL  IOSB, STATUS                                ;5136
                                  19          56 E8 00196          BLBS    STATUS, 13$
                                              56 DD 00199 12$:     PUSHL   STATUS
                                  00000000'   EF 9F 0019B          PUSHAB  RSA_DESC
                                              01 DD 001A1          PUSHL   #1
                                  00000000G   8F DD 001A3          PUSHL   #BACKUP$_READERR+2
                00000000G         00          04 FB 001A9          CALLS   #4, LIB$SIGNAL
                                              31 11 001B0          BRB     18$
                                  52     008C CE 9E 001B2 13$:     MOVAB   BUFFER, R2                                  ;5138
                                  54     FEE0 CD 9E 001B7          MOVAB   BUFFER+480, R4
                                              20 11 001BC          BRB     17$
                                  1A          62 E9 001BE 14$:     BLBC    (P), 16$                                    ;5148
                                        04    A2 D5 001C1          TSTL    4(P)                                        ;5151
                                              08 12 001C4          BNEQ    15$
                                  58     0C   A2 D0 001C6          MOVL    12(P), DQF_DEFAULT_PERM                     ;5154
                                  57     10   A2 D0 001CA          MOVL    16(P), DQF_DEFAULT_OVER                     ;5155
                                        04    A2 DD 001CE 15$:     PUSHL   4(P)                                        ;5157
                              E38E      CF    01 FB 001D1          CALLS   #1, DQF_FIND_UIC
                                10      A0    0C A2 7D 001D6       MOVQ    12(P), T6(Q)                                ;5158
                                  52          20 C0 001DB 16$:     ADDL2   #32, P                                      ;5138
                                  54          52 D1 001DE 17$:     CMPL    P, R4
                                              DB 1B 001E1          BLEQU   14$
                    80                    53    55 F2 001E3 18$:     AOBLSS  R5, VBN, 11$                                ;5121
                    51          20        AE    10 9C 001E7          ROTL    #16, FAT+4, R1                              ;5168
                    50 00000000'          EF    10 C7 001EC          DIVL3   #16, DQF_COUNT, R0
                                  50          51 D1 001F4          CMPL    R1, R0
                                              28 1E 001F7          BGEQU   19$
                                  00000000G   8F DD 001F9          PUSHL   #BACKUP$_QUOTAFILE                          ;5171
                        00000000G 00          01 FB 001FF          CALLS   #1, LIB$SIGNAL
                                              7E 7C 00206          CLRQ    -(SP)                                       ;5174
                                              7E 7C 00208          CLRQ    -(SP)
                                              7E 7C 0020A          CLRQ    -(SP)
                                              7E 7C 0020C          CLRQ    -(SP)
                                  7E          34 7D 0020E          MOVQ    #52, -(SP)
                                  0002FFFF    8F DD 00211          PUSHL   #196607
                                              7E D4 00217          CLRL    -(SP)
                00000000G         00          0C FB 00219          CALLS   #12, STA_QIOW
                                              04 00220          RET                                                 ;5170
0200    8F          00            6E          00 2C 00221 19$:     MOVC5   #0, (SP), #0, #512, BUFFER                  ;5181
```

```
                                 008C   CE      00228
                            5A   008C   CE  9E 0022B          MOVAB   BUFFER, DQF_RECORD                          ;5182
                            5B          5A  D0 00230          MOVL    DQF_RECORD, DQF_BUFFER
                            59          01  D0 00233          MOVL    #1, DQF_VBN                                 ;5183
                            50 00000000' EF  D0 00236          MOVL    DQF_ROOT, R0                                ;5184
                                        07  13 0023D          BEQL    20$
                                        50  DD 0023F          PUSHL   R0
                        E37A    CF          01  FB 00241          CALLS   #1, DQF_WRITE_ENTRY
                            5B          5A  D1 00246 20$:      CMPL    DQF_RECORD, DQF_BUFFER                      ;5189
                                        49  13 00249          BEQL    22$
                                        7E  7C 0024B          CLRQ    -(SP)                                       ;5198
                                        7E  D4 0024D          CLRL    -(SP)
                                        59  DD 0024F          PUSHL   DQF_VBN
                            7E      0200    8F  3C 00251          MOVZWL  #512, -(SP)
                                        5B  DD 00256          PUSHL   DQF_BUFFER
                                        7E  7C 00258          CLRQ    -(SP)
                                     5C     AE  9F 0025A          PUSHAB  IOSB
                                        30  DD 0025D          PUSHL   #48
                                 0002FFFF   8F  DD 0025F          PUSHL   #196607
                                        7E  D4 00265          CLRL    -(SP)
                    00000000G   00          0C  FB 00267          CALLS   #12, STA_QIOW
                            56          50  D0 0026E          MOVL    R0, STATUS
                                        59  D6 00271          INCL    DQF_VBN                                     ;5199
                            07          56  E9 00273          BLBC    STATUS, 21$                                 ;5200
                            56      3C      AE  3C 00276          MOVZWL  IOSB, STATUS
                            17          56  E8 0027A          BLBS    STATUS, 22$                                 ;5201
                                        56  DD 0027D 21$:      PUSHL   STATUS                                      ;5203
                                 00000000'  EF  9F 0027F          PUSHAB  RSA_DESC
                                        01  DD 00285          PUSHL   #1
                                 00000000G  8F  DD 00287          PUSHL   #BACKUP$_WRITEERR+2
                    00000000G   00          04  FB 0028D          CALLS   #4, LIB$SIGNAL
                24  AE          59          10  9C 00294 22$:      ROTL    #16, DQF_VBN, FAT+8                         ;5209
                                        7E  D4 00299          CLRL    -(SP)                                       ;5214
                                     14     AE  9F 0029B          PUSHAB  ATR_DESC
                                        7E  7C 0029E          CLRQ    -(SP)
                                        7E  7C 002A0          CLRQ    -(SP)
                                        7E  7C 002A2          CLRQ    -(SP)
                                     5C     AE  9F 002A4          PUSHAB  IOSB
                                        34  DD 002A7          PUSHL   #52
                                 0002FFFF   8F  DD 002A9          PUSHL   #196607
                                        7E  D4 002AF          CLRL    -(SP)
                    00000000G   00          0C  FB 002B1          CALLS   #12, STA_QIOW
                            56          50  D0 002B8          MOVL    R0, STATUS
                            07          56  E9 002BB          BLBC    STATUS, 23$                                 ;5215
                            56      3C      AE  3C 002BE          MOVZWL  IOSB, STATUS
                            17          56  E8 002C2          BLBS    STATUS, 24$                                 ;5216
                                        56  DD 002C5 23$:      PUSHL   STATUS                                      ;5218
                                 00000000'  EF  9F 002C7          PUSHAB  RSA_DESC
                                        01  DD 002CD          PUSHL   #1
                                 00000000G  8F  DD 002CF          PUSHL   #BACKUP$_CLOSEOUT+2
                    00000000G   00          04  FB 002D5          CALLS   #4, LIB$SIGNAL
                            50 00000000' EF  D0 002DC 24$:      MOVL    CURRENT_MTL, R0                             ;5226
                            01      1F      A0  91 002E3          CMPB    31(R0), #1
                                        03  1A 002E7          BGTRU   26$
                                     0198   31  002E9 25$:      BRW     36$
                                        6C  95 002EC 26$:      TSTB    (AP)                                        ;5227
                                        F9  12 002EE          BNEQ    25$
```

```
                    00000000'  EF       34   A0  D0 002F0          MOVL    52(R0), CURRENT_VCB                     ; 5239
                    00000000'  EF       FF   8F  9A 002F8          MOVZBL  #255, RSA_DESC                          ; 5240
                 7E 00000000'  EF            20  C1 00300          ADDL3   #32, CURRENT_VCB, -(SP)                 ; 5245
                               00000000'  EF  9F  00308          PUSHAB  RSA_DESC                                ;
                               00000000'  EF  9F  0030E          PUSHAB  RSA_DESC                                ;
                                     FC80 CF  9F  00314          PUSHAB  P.AAK                                   ;
                    00000000G  00            04  FB  00318          CALLS   #4, SYS$FAO                             ;
                          4C   AE      0101  8F  3C  0031F          MOVZWL  #257, FIB                               ; 5246
                          50   AE  00060006  8F  D0  00325          MOVL    #393222, FIB+4                          ; 5247
                          54   AE            01  B0  0032D          MOVW    #1, FIB+8                               ; 5249
                                            7E  7C  00331          CLRQ    -(SP)                                   ; 5254
                                            7E  7C  00333          CLRQ    -(SP)                                   ;
                                            7E  D4  00335          CLRL    -(SP)                                   ;
                                       58   AE  9F  00337          PUSHAB  FIB_DESC                                ;
                                            7E  7C  0033A          CLRQ    -(SP)                                   ;
                                       5C   AE  9F  0033C          PUSHAB  IOSB                                    ;
                                    7E      72  8F  9A  0033F          MOVZBL  #114, -(SP)                             ;
                                        0002FFFF  8F  DD  00343          PUSHL   #196607                                 ;
                                            7E  D4  00349          CLRL    -(SP)                                   ;
                    00000000G  00            0C  FB  0034B          CALLS   #12, STA_QIOW                           ;
                               56            50  D0  00352          MOVL    R0, STATUS                              ;
                               07            56  E9  00355          BLBC    STATUS, 27$                             ; 5255
                               56       3C   AE  3C  00358          MOVZWL  IOSB, STATUS                            ;
                               17            56  E8  0035C          BLBS    STATUS, 28$                             ; 5256
                                            56  DD  0035F 27$:      PUSHL   STATUS                                  ; 5258
                               00000000'  EF  9F  00361          PUSHAB  RSA_DESC                                ;
                                            01  DD  00367          PUSHL   #1                                      ;
                               00000000G  8F  DD  00369          PUSHL   #BACKUP$_OPENOUT+2                      ;
                    00000000G  00            04  FB  0036F          CALLS   #4, LIB$SIGNAL                          ;
    0200    8F           00 00000000'  EF            0C  2C  00376 28$:    MOVC5   #12, COM_O_STRUCNAME, #0, #512, BUFFER  ; 5263
                                     008C CE      00381                                                           ;
                               58       40  8F  9A  00384          MOVZBL  #64, P                                  ; 5264
                               59            01  D0  00388          MOVL    #1, VBN                                 ; 5265
                               50 00000000'  EF  D0  0038B          MOVL    CURRENT_MTL, R0                         ; 5268
                               5A       1F   A0  9A  00392          MOVZBL  31(R0), R10                             ;
                                            57  D4  00396          CLRL    RVN                                     ;
                                          0080  31  00398          BRW     33$                                     ;
                    00000200   8F            58  D1  0039B 29$:    CMPL    P, #512                                 ; 5273
                                            57  1F  003A2          BLSSU   32$                                     ;
                                            7E  7C  003A4          CLRQ    -(SP)                                   ; 5282
                                            7E  D4  003A6          CLRL    -(SP)                                   ;
                                            59  DD  003A8          PUSHL   VBN                                     ;
                                    7E      0200  8F  3C  003AA          MOVZWL  #512, -(SP)                             ;
                                        00A0  CE  9F  003AF          PUSHAB  BUFFER                                  ;
                                            7E  7C  003B3          CLRQ    -(SP)                                   ;
                                       5C   AE  9F  003B5          PUSHAB  IOSB                                    ;
                                            30  DD  003B8          PUSHL   #48                                     ;
                                        0002FFFF  8F  DD  003BA          PUSHL   #196607                                 ;
                                            7E  D4  003C0          CLRL    -(SP)                                   ;
                    00000000G  00            0C  FB  003C2          CALLS   #12, STA_QIOW                           ;
                               56            50  D0  003C9          MOVL    R0, STATUS                              ;
                               07            56  E9  003CC          BLBC    STATUS, 30$                             ; 5283
                               56       3C   AE  3C  003CF          MOVZWL  IOSB, STATUS                            ;
                               17            56  E8  003D3          BLBS    STATUS, 31$                             ; 5284
                                            56  DD  003D6 30$:      PUSHL   STATUS                                  ; 5286
                               00000000'  EF  9F  003D8          PUSHAB  RSA_DESC                                ;
                                            01  DD  003DE          PUSHL   #1                                      ;
```

```
                                                00000000G  8F  DD 003E0          PUSHL   #BACKUP$_WRITEERR+2                          ;
                                    00000000G   00         04 FB 003E6          CALLS   #4, LIB$SIGNAL                              ;
0200    8F                  00                  6E         00 2C 003ED 31$:     MOVC5   #0, (SP), #0, #512, BUFFER                  ; 5291
                                                       008C CE      003F4                                                           ;
                                                            58 D4 003F7          CLRL    P                                           ; 5292
                                                            59 D6 003F9          INCL    VBN                                         ; 5293
                                                51 00000000' EF DO 003FB 32$:     MOVL    CURRENT_MTL, R1                             ; 5301
                                                50       30 A1 9A 00402          MOVZBL  48(R1), R0                                  ;
                                        50                  57 C3 00406          SUBL3   R0, RVN, R0                                 ;
                                                50       34 A140 D0 0040A        MOVL    52(R1)[R0], R0                              ;
                                    008C CE48       38      A0 0C 28 0040F       MOVC3   #12, 56(R0), BUFFER[P]                      ; 5302
                                                58       40 A8 9E 00417          MOVAB   64(R8), P                                   ; 5305
                        FF7A                    57          01 5A F1 0041B 33$:  ACBL    R10, #1, RVN, 29$                           ; 5268
                                                            7E 7C 00421          CLRQ    -(SP)                                       ; 5317
                                                            7E D4 00423          CLRL    -(SP)                                       ;
                                                            59 DD 00425          PUSHL   VBN                                         ;
                                                7E       0200 8F 3C 00427        MOVZWL  #512, -(SP)                                 ;
                                                         00A0 CE 9F 0042C        PUSHAB  BUFFER                                      ;
                                                            7E 7C 00430          CLRQ    -(SP)                                       ;
                                                            5C AE 9F 00432       PUSHAB  IOSB                                        ;
                                                            30 DD 00435          PUSHL   #48                                         ;
                                                00020FFF    8F DD 00437          PUSHL   #196607                                     ;
                                                            7E D4 0043D          CLRL    -(SP)                                       ;
                                    00000000G   00          0C FB 0043F          CALLS   #12, STA_QIOW                               ;
                                                56          50 D0 00446          MOVL    R0, STATUS                                  ;
                                                07          56 E9 00449          BLBC    STATUS, 34$                                 ; 5318
                                                56       3C AE 3C 0044C          MOVZWL  IOSB, STATUS                                ;
                                                17          56 E8 00450          BLBS    STATUS, 35$                                 ; 5319
                                                            56 DD 00453 34$:     PUSHL   STATUS                                      ; 5321
                                                00000000'   EF 9F 00455          PUSHAB  RSA_DESC                                    ;
                                                            01 DD 0045B          PUSHL   #1                                          ;
                                                00000000G   8F DD 0045D          PUSHL   #BACKUP$_WRITEERR+2                         ;
                                    00000000G   00          04 FB 00463          CALLS   #4, LIB$SIGNAL                              ;
                                                            7E 7C 0046A 35$:     CLRQ    -(SP)                                       ; 5328
                                                            7E 7C 0046C          CLRQ    -(SP)                                       ;
                                                            7E 7C 0046E          CLRQ    -(SP)                                       ;
                                                            7E 7C 00470          CLRQ    -(SP)                                       ;
                                                7E          34 7D 00472          MOVQ    #52, -(SP)                                  ;
                                                0002FFFF    8F DD 00475          PUSHL   #196607                                     ;
                                                            7E D4 0047B          CLRL    -(SP)                                       ;
                                    00000000G   00          0C FB 0047D          CALLS   #12, STA_QIOW                               ;
                                                50 00000000' EF DO 00484 36$:     MOVL    CURRENT_MTL, R0                             ; 5336
                                    00000000'   EF       34 A0 D0 0048B          MOVL    52(R0), CURRENT_VCB                         ;
                                                02          6C 91 00493          CMPB    (AP), #2                                    ; 5337
                                                            03 1E 00496          BGEQU   38$                                         ;
                                                         016C 31 00498 37$:      BRW     46$                                         ;
                                                01       04 AC D1 0049B 38$:     CMPL    P_RVN, #1                                   ; 5338
                                                            F7 12 0049F          BNEQ    37$                                         ;
                                                F3       08 AC E9 004A1          BLBC    CONTINUE, 37$                               ; 5339
                                                51 00000000' EF DO 004A5         MOVL    CURRENT_VCB, R1                             ; 5340
                        E7          07          A1          04 E1 004AC          BBC     #4, 7(R1), 37$                              ;
                                    08          A0 00000000' FF DO 004B1         MOVL    @CURRENT_VCB, 8(R0)                         ; 5349
                                00000000'       EF       8F 9A 004B9             MOVZBL  #255, RSA_DESC                              ; 5350
                        7E      00000000'       EF          20 C1 004C1          ADDL3   #32, CURRENT_VCB, -(SP)                     ; 5355
                                                00000000'   EF 9F 004C9          PUSHAB  RSA_DESC                                    ;
                                                00000000'   EF 9F 004CF          PUSHAB  RSA_DESC                                    ;
                                                    FADF    CF 9F 004D5          PUSHAB  P.AAM                                       ;
```

```
                00000000G  00              04  FB 004D9              CALLS   #4, SYS$FAO                                     ;
                           50 00000000'     EF  D0 004E0              MOVL    CURRENT_MTL, R0                                 ; 5356
00000000'                  EF       08      A0  D0 004E7              MOVL    8(R0), CURRENT_WCB                              ;
                           50 00000000'     EF  D0 004EF              MOVL    CURRENT_VCB, R0                                 ; 5362
                           58       04      A0  3C 004F6              MOVZWL  4(R0), R8                                       ;
                           58               03  C4 004FA              MULL2   #3, R8                                          ;
                           57               01  D0 004FD              MOVL    #1, VBN                                         ;
                                          00F4  31 00500              BRW     45$                                             ;
                                            57  DD 00503  39$:        PUSHL   VBN                                             ; 5374
                           7E       0200    8F  3C 00505              MOVZWL  #512, -(SP)                                     ; 5365
                                    0094    CE  9F 0050A              PUSHAB  BUFFER                                          ;
                                            7E  7C 0050E              CLRQ    -(SP)                                           ;
                                       50   AE  9F 00510              PUSHAB  IOSB                                            ;
                                            31  DD 00513              PUSHL   #49                                             ;
                                            7E  7C 00515              CLRQ    -(SP)                                           ;
                        EB34   CF           09  FB 00517              CALLS   #9, R_W_VIRTUAL                                 ;
                               56           50  D0 0051C              MOVL    R0, STATUS                                      ;
                                            7E  D4 0051F              CLRL    -(SP)                                           ; 5375
                00000000G  00               01  FB 00521              CALLS   #1, SYS$WAITFR                                  ;
                           07               56  E9 00528              BLBC    STATUS, 40$                                     ; 5376
                           56       3C      AE  3C 0052B              MOVZWL  IOSB, STATUS                                    ;
                           17               56  E8 0052F              BLBS    STATUS, 41$                                     ; 5377
                                            56  DD 00532  40$:        PUSHL   STATUS                                          ; 5379
                                00000000'   EF  9F 00534              PUSHAB  RSA_DESC                                        ;
                                            01  DD 0053A              PUSHL   #1                                              ;
                                00000000G   8F  DD 0053C              PUSHL   #BACKUP$_READERR+2                              ;
                00000000G  00               04  FB 00542              CALLS   #4, LIB$SIGNAL                                  ;
                                            3A  DD 00549  41$:        PUSHL   #58                                             ; 5382
                                      0090  CE  9F 0054B              PUSHAB  BUFFER                                          ;
                00000000G  00               02  FB 0054F              CALLS   #2, CHECKSUM2                                   ;
                           13               50  E9 00556              BLBC    R0, 42$                                         ;
                           7E       01FE    8F  3C 00559              MOVZWL  #510, -(SP)                                     ; 5383
                                    0090    CE  9F 0055E              PUSHAB  BUFFER                                          ;
                00000000G  00               02  FB 00562              CALLS   #2, CHECKSUM2                                   ;
                           15               50  E8 00569              BLBS    R0, 43$                                         ;
                                00000000'   EF  9F 0056C  42$:        PUSHAB  RSA_DESC                                        ; 5384
                                            01  DD 00572              PUSHL   #1                                              ;
                                00000000G   8F  DD 00574              PUSHL   #BACKUP$_INVHOMBLK                              ;
                00000000G  00               03  FB 0057A              CALLS   #3, LIB$SIGNAL                                  ;
                              00B2  CE      01  B0 00581  43$:        MOVW    #1, BUFFER+38                                   ; 5390
                           50 00000000'     EF  D0 00586              MOVL    CURRENT_MTL, R0                                 ; 5391
FECC    CD            24   A0              0C  28 0058D              MOVC3   #12, 36(R0), BUFFER+460                         ;
                                            3A  DD 00594              PUSHL   #58                                             ; 5392
                                      0090  CE  9F 00596              PUSHAB  BUFFER                                          ;
                00000000G  00               02  FB 0059A              CALLS   #2, CHECKSUM2                                   ;
                           7E       01FE    8F  3C 005A1              MOVZWL  #510, -(SP)                                     ; 5393
                                    0090    CE  9F 005A6              PUSHAB  BUFFER                                          ;
                00000000G  00               02  FB 005AA              CALLS   #2, CHECKSUM2                                   ;
                                            57  DD 005B1              PUSHL   VBN                                             ; 5407
                           7E       0200    8F  3C 005B3              MOVZWL  #512, -(SP)                                     ; 5398
                                    0094    CE  9F 005B8              PUSHAB  BUFFER                                          ;
                                            7E  7C 005BC              CLRQ    -(SP)                                           ;
                                       50   AE  9F 005BE              PUSHAB  IOSB                                            ;
                                            30  DD 005C1              PUSHL   #48                                             ;
                                            7E  7C 005C3              CLRQ    -(SP)                                           ;
                           EA86   CF           09  FB 005C5              CALLS   #9, R_W_VIRTUAL                                 ;
                                  56           50  D0 005CA              MOVL    R0, STATUS                                      ;
```

```
                                                                7E  D4 005CD            CLRL    -(SP)                                   ; 5408
                                                 00000000G      00                   01  FB 005CF            CALLS   #1, SYS$WAITFR
                                                                07                   56  E9 005D6            BLBC    STATUS, 44$             ; 5409
                                                                56          3C       AE  3C 005D9            MOVZWL  IOSB, STATUS
                                                                17                   56  E8 005DD            BLBS    STATUS, 45$             ; 5410
                                                                                     56  DD 005E0 44$:       PUSHL   STATUS                  ; 5412
                                                                    00000000'        EF  9F 005E2            PUSHAB  RSA_DESC
                                                                                     01  DD 005E8            PUSHL   #1
                                                                    00000000G        8F  DD 005EA            PUSHL   #BACKUP$_WRITEERR+2
                                                 00000000G      00                   04  FB 005F0            CALLS   #4, LIB$SIGNAL
                                        FF06             57     01                   58  F1 005F7 45$:       ACBL    R8, #1, VBN, 39$        ; 5362
                                                                50 00000000'        EF  D0 005FD            MOVL    CURRENT_MTL, R0         ; 5416
                                                                          08        A0  D4 00604            CLRL    8(R0)
                                                                                    6C  95 00607 46$:       TSTB    (AP)                    ; 5423
                                                                                    0D  12 00609            BNEQ    47$
                                                                50 00000000'        EF  D0 0060B            MOVL    CURRENT_MTL, R0         ; 5424
                                                                59          30      A0  9A 00612            MOVZBL  48(R0), R9
                                                                                    04  11 00616            BRB     48$
                                                                59          04      AC  D0 00618 47$:       MOVL    P_RVN, R9               ; 5425
                                                                                    6C  95 0061C 48$:       TSTB    (AP)                    ; 5426
                                                                                    18  12 0061E            BNEQ    49$
                                                                50 00000000'        EF  D0 00620            MOVL    CURRENT_MTL, R0         ; 5427
                                                                51          30      A0  9A 00627            MOVZBL  48(R0), R1
                                                                50          1F      A0  9A 0062B            MOVZBL  31(R0), R0
                                                                50                  51  C0 0062F            ADDL2   R1, R0
                                                       04       AE                  70  9E 00632            MOVAB   -(R0), 4(SP)
                                                                                    05  11 00636            BRB     50$
                                                       04       AE          04      AC  D0 00638 49$:       MOVL    P_RVN, 4(SP)            ; 5428
                                                                                    59  D7 0063D 50$:       DECL    RVN                     ; 5422
                                                                                01D8 31  00063F            BRW     66$
                                                                51 00000000'        EF  D0 00642 51$:       MOVL    CURRENT_MTL, R1         ; 5438
                                                                50          30      A1  9A 00649            MOVZBL  48(R1), R0
                                                 50                 59              50  C3 0064D            SUBL3   R0, RVN, R0
                                                                5A          34 A140     D0  00651            MOVL    52(R1)[R0], VCB
                                                    00000000'   EF                  5A  D0 00656            MOVL    VCB, CURRENT_VCB
                                                    00000000'   EF          FF      8F  9A 0065D            MOVZBL  #255, RSA_DESC          ; 5439
                                                                            20      AA  9F 00665            PUSHAB  32(VCB)                 ; 5444
                                                                    00000000'       EF  9F 00668            PUSHAB  RSA_DESC
                                                                    00000000'       EF  9F 0066E            PUSHAB  RSA_DESC
                                                                        F960        CF  9F 00674            PUSHAB  P.AAO
                                                    00000000G   00                  04  FB 00678            CALLS   #4, SYS$FAO
                                                                                    59  DD 0067F            PUSHL   RVN                     ; 5445
                                                        E01F    CF                  01  FB 00681            CALLS   #1, SWITCH_VOLUME
                                                          08    AE                  50  D0 00686            MOVL    R0, CHANNEL             ; 5456
                                                                                    7E  7C 0068A            CLRQ    -(SP)
                                                                                    7E  D4 0068C            CLRL    -(SP)
                                                                            14      AA  DD 0068E            PUSHL   20(VCB)
                                                 7E                 1C      AA          08  C7 00691        DIVL3   #8, 28(VCB), -(SP)
                                                                            10      AA  DD 00696            PUSHL   16(VCB)
                                                                                    7E  7C 00699            CLRQ    -(SP)
                                                                            5C      AE  9F 0069B            PUSHAB  IOSB
                                                                            20      DD  DD 0069E            PUSHL   #32
                                                                            30      AE  DD 006A0            PUSHL   CHANNEL
                                                                                    7E  D4 006A3            CLRL    -(SP)
                                                    00000000G   00                  0C  FB 006A5            CALLS   #12, SYS$QIOW
                                                                56                  50  D0 006AC            MOVL    R0, STATUS
                                                                07                  56  E9 006AF            BLBC    STATUS, 52$             ; 5457
```

```
                                56       3C    AE  3C 006B2          MOVZWL  IOSB, STATUS                                    ;
                                17             56  E8 006B6          BLBS    STATUS, 53$                                     ; 5458
                                               56  DD 006B9 52$:     PUSHL   STATUS                                          ; 5460
                                   00000000'   EF  9F 006BB          PUSHAB  RSA_DESC                                        ;
                                               01  DD 006C1          PUSHL   #1                                              ;
                                   00000000G   8F  DD 006C3          PUSHL   #BACKUP$_WRITEERR+2                             ;
                          00000000G  00        04  FB 006C9          CALLS   #4, LIB$SIGNAL                                  ;
                                5B             01  CE 006D0 53$:     MNEGL   #1, OFFSET                                      ; 5465
                                57       28    AA  D0 006D3          MOVL    40(VCB), ACB                                    ; 5466
                          00000000'  EF        FF  8F 006D7          MOVZBL  #255, RSA_DESC                                  ; 5467
                                           20  AA  9F 006DF          PUSHAB  32(VCB)                                         ; 5472
                                   00000000'   EF  9F 006E2          PUSHAB  RSA_DESC                                        ;
                                   00000000'   EF  9F 006E8          PUSHAB  RSA_DESC                                        ;
                                       F906    CF  9F 006EE          PUSHAB  P.AAQ                                           ;
                          00000000G  00        04  FB 006F2          CALLS   #4, SYS$FAO                                     ;
                                57       28    BA  0F 006F9 54$:     REMQUE  @40(VCB), ACB                                   ; 5477
                                               03  1C 006FD          BVC     55$                                             ;
                                             009F  31 006FF          BRW     62$                                             ;
                                58       04    AA  3C 00702 55$:     MOVZWL  4(VCB), R8                                      ; 5484
                    58      0C   A7             58  C7 00706          DIVL3   R8, 12(ACB), R8                                 ;
                                50       04    AA  3C 0070B          MOVZWL  4(VCB), R0                                      ; 5485
                    50      08   A7             50  C7 0070F          DIVL3   R0, 8(ACB), R0                                  ;
                                6E       FF A048  9E 00714          MOVAB   -1(R0)[R8], (SP)                                ;
                                               73  11 00719          BRB     61$                                             ; 5483
          5B        58                   14       0C  ED 0071B 56$:     CMPZV   #12, #20, N, OFFSET                             ; 5493
                                                    5F  15 00720          BLEQ    59$                                             ;
                          FFFFFFFF   8F              5B  D1 00722          CMPL    OFFSET, #-1                                     ; 5496
                                                    48  13 00729          BEQL    58$                                             ;
                                                    7E  7C 0072B          CLRQ    -(SP)                                           ; 5505
                                                    7E  D4 0072D          CLRL    -(SP)                                           ;
                                              0C BA4B   9F 0072F          PUSHAB  @12(VCB)[OFFSET]                                ;
                                   7E       0200   8F  3C 00733          MOVZWL  #512, -(SP)                                     ;
                                            00A0   CE  9F 00738          PUSHAB  BUFFER                                          ;
                                                   7E  7C 0073C          CLRQ    -(SP)                                           ;
                                              5C   AE  9F 0073E          PUSHAB  IOSB                                            ;
                                              20   DD  00741             PUSHL   #32                                             ;
                                              30   AE  DD 00743          PUSHL   CHANNEL                                         ;
                                                   7E  D4 00746          CLRL    -(SP)                                           ;
                          00000000G  00            0C  FB 00748          CALLS   #12, SYS$QIOW                                   ;
                                56                 50  D0 0074F          MOVL    R0, STATUS                                      ;
                                07                 56  E9 00752          BLBC    STATUS, 57$                                     ; 5506
                                56       3C        AE  3C 00755          MOVZWL  IOSB, STATUS                                    ;
                                17                 56  E8 00759          BLBS    STATUS, 58$                                     ; 5507
                                                   56  DD 0075C 57$:     PUSHL   STATUS                                          ; 5509
                                   00000000'       EF  9F 0075E          PUSHAB  RSA_DESC                                        ;
                                                   01  DD 00764          PUSHL   #1                                              ;
                                   00000000G       8F  DD 00766          PUSHL   #BACKUP$_WRITEERR+2                             ;
                          00000000G  00            04  FB 0076C          CALLS   #4, LIB$SIGNAL                                  ;
0200      8F        00                   6E            00  2C 00773 58$:     MOVC5   #0, (SP), #0, #512, BUFFER                      ; 5511
                                                  008C   CE     0077A                                                              ;
                                                   5B    D6 0077D          INCL    OFFSET                                          ; 5512
                                                   9A    11 0077F          BRB     56$                                             ; 5493
          50        58                   0C             00  EF 00781 59$:     EXTZV   #0, #12, N, R0                                  ; 5518
                    00       008C        CE             50  E2 00786          BBSS    R0, BUFFER, 60$                                 ;
                                                        58  D6 0078C 60$:     INCL    N                                               ; 5483
                                6E                      58  D1 0078E 61$:     CMPL    N, (SP)                                         ;
                                                        88  1B 00791          BLEQU   56$                                             ;
```

```
                                                  57  DD 00793          PUSHL   ACB                                             ; 5524
                                                  10  DD 00795          PUSHL   #16
                          00000000G   00          02  FB 00797          CALLS   #2, FREE_VM
                                                FF58  31 0079E          BRW     54$                                             ; 5477
        5B      34    AA              10          00  ED 007A1 62$:     CMPZV   #0, #16, 52(VCB), OFFSET                        ; 5530
                                                  5F  15 007A7          BLEQ    65$                                             ; 5533
                          FFFFFFFF    8F          5B  D1 007A9          CMPL    OFFSET, #-1
                                                  48  13 007B0          BEQL    64$                                             ; 5542
                                                  7E  7C 007B2          CLRQ    -(SP)
                                                  7E  D4 007B4          CLRL    -(SP)
                                               ^C BA4B  9F 007B6          PUSHAB  @12(VCB)[OFFSET]
                                7E      0200      8F  3C 007BA          MOVZWL  #512, -(SP)
                                        00A0      CE  9F 007BF          PUSHAB  BUFFER
                                                  7E  7C 007C3          CLRQ    -(SP)
                                             5C   AE  9F 007C5          PUSHAB  IOSB
                                                  20  DD 007C8          PUSHL   #32
                                             30   AE  DD 007CA          PUSHL   CHANNEL
                                                  7E  D4 007CD          CLRL    -(SP)
                          00000000G   00          0C  FB 007CF          CALLS   #12, SYS$QIOW
                                      56          50  D0 007D6          MOVL    R0, STATUS
                                      07          56  E9 007D9          BLBC    STATUS, 63$                                     ; 5543
                                      56     3C   AE  3C 007DC          MOVZWL  IOSB, STATUS
                                      17          56  E8 007E0          BLBS    STATUS, 64$                                     ; 5544
                                                  56  DD 007E3 63$:     PUSHL   STATUS                                          ; 5546
                                         00000000' EF  9F 007E5          PUSHAB  RSA_DESC
                                                  01  DD 007EB          PUSHL   #1
                                         00000000G 8F  DD 007ED          PUSHL   #BACKUP$_WRITEERR+2
                          00000000G   00          04  FB 007F3          CALLS   #4, LIB$SIGNAL
    0200    8F            00        6E          00  2C 007FA 64$:     MOVC5   #0, (SP), #0, #512, BUFFER                      ; 5548
                                            008C  CE     00801
                                                  5B  D6 00804          INCL    OFFSET                                          ; 5549
                                                  99  11 00806          BRB     62$                                             ; 5530
                                             10   AA  DD 00808 65$:     PUSHL   16(VCB)                                         ; 5555
                          7E        1C    AA              08  C7 0080B          DIVL3   #8, 28(VCB), -(SP)
                              00000000G   00              02  FB 00810          CALLS   #2, FREE_VM
                                             10   AA  D4 00817          CLRL    16(VCB)                                         ; 5556
        FE21              59          01        04  AE  F1 0081A 66$:     ACBL    4(SP), #1, RVN, 51$                             ; 5422
                                      50 00000000' EF  D0 00821          MOVL    CURRENT_MTL, R0                                 ; 5562
                                   08   A0        0C  AE  D0 00828          MOVL    SAVE_WCB, 8(R0)
                                                  04  0082D          RET                                                     ; 5563

; Routine Size:  2094 bytes,    Routine Base:  CODE + 1A9C
```

```
;  4040          5564  1 %SBTTL 'STA_DISMOUNT - dismount input volume'
;  4041          5565  1 GLOBAL ROUTINE STA_DISMOUNT (P_RVN) : NOVALUE=
;  4042          5566  1
;  4043          5567  1 !++
;  4044          5568  1 !
;  4045          5569  1 ! FUNCTIONAL DESCRIPTION:
;  4046          5570  1 !       This routine is called after reading a volume of sequential
;  4047          5571  1 !       disk, to clean up the disk dependent data structures.
;  4048          5572  1 !
;  4049          5573  1 ! INPUT PARAMETERS:
;  4050          5574  1 !       P_RVN (optional): if present, specifies RVN to dismount
;  4051          5575  1 !                         if absent, dismount entire set
;  4052          5576  1 !
;  4053          5577  1 ! IMPLICIT INPUTS:
;  4054          5578  1 !       INPUT_MTL       - Pointer to MTL for input volume set.
;  4055          5579  1 !
;  4056          5580  1 ! INPUT PARAMETERS:
;  4057          5581  1 !       NONE
;  4058          5582  1 !
;  4059          5583  1 ! IMPLICIT INPUTS:
;  4060          5584  1 !       NONE
;  4061          5585  1 !
;  4062          5586  1 ! ROUTINE VALUE:
;  4063          5587  1 !       NONE
;  4064          5588  1 !
;  4065          5589  1 ! SIDE EFFECTS:
;  4066          5590  1 !       NONE
;  4067          5591  1 !
;  4068          5592  1 !--
;  4069          5593  1
;  4070          5594  2 BEGIN
;  4071          5595  2 LOCAL
;  4072          5596  2         ACB             : REF BBLOCK;   ! allocation control block
;  4073          5597  2
;  4074          5598  2
;  4075          5599  2 ! Free the index file window.
;  4076          5600  2 !
;  4077          5601  2 CURRENT_VCB = .CURRENT_MTL[MTL_VCB(.P_RVN-.CURRENT_MTL[MTL_RVN_BASE])];
;  4078          5602  2 DELETE_WINDOW (.CURRENT_VCB[VCB_INDEXF]);
;  4079          5603  2 CURRENT_VCB[VCB_INDEXF] = 0;
;  4080          5604  2
;  4081          5605  2 ! Free the index file bitmap buffers.
;  4082          5606  2 !
;  4083          5607  2 IF .CURRENT_VCB[VCB_IMAP] NEQ 0
;  4084          5608  2 THEN
;  4085          5609  3     BEGIN
;  4086          5610  3     FREE_VM(.CURRENT_VCB[VCB_MAXFILIDX]/8, .CURRENT_VCB[VCB_IMAP]);
;  4087          5611  3     CURRENT_VCB[VCB_IMAP] = 0;
;  4088          5612  2     END;
;  4089          5613  2
;  4090          5614  2 ! Free any ACB's lying around.
;  4091          5615  2 !
;  4092          5616  2
;  4093          5617  2 WHILE NOT REMQUE (.CURRENT_VCB[VCB_ACB_FLINK], ACB)
;  4094          5618  2 DO
;  4095          5619  3     BEGIN
;  4096          5620  3     FREE_VM(ACB_S_ENTRY, .ACB);
```

```
;  4097          5621  2     END;
;  4098          5622  2
;  4099          5623  2 ! Clean up the VCB and MTL.
;  4100          5624  2 !
;  4101          5625  2
;  4102          5626  2 CURRENT_VCB[VCB_ODS_2] = 0;
;  4103          5627  2 CURRENT_VCB[VCB_INIT_DONE] = 0;
;  4104          5628  2 CURRENT_VCB[VCB_SAVESET] = 0;
;  4105          5629  2 CURRENT_VCB[VCB_NOTVOLSET] = 0;
;  4106          5630  2
;  4107          5631  2 DELETE_WINDOW (.CURRENT_MTL[MTL_WINDOW]);
;  4108          5632  2 CURRENT_MTL[MTL_WINDOW] = 0;
;  4109          5633  2
;  4110          5634  1 END;


                                                001C 00000          .ENTRY  STA_DISMOUNT, Save R2,R3,R4                     ; 5565
                                    54 00000000G   00   9E 00002    MOVAB   FREE_VM, R4                                     ;
                                    53 00000000'   EF   9E 00009    MOVAB   CURRENT_VCB, R3                                 ;
                                    51         FC   A3   D0 00010    MOVL    CURRENT_MTL, R1                         ; 5601
                                    50         30   A1   9A 00014    MOVZBL  48(R1), R0                                      ;
                                 50       04   AC        50   C3 00018    SUBL3   R0, P_RVN, R0                           ;
                                    63         34 A140   D0 0001D    MOVL    52(R1)[R0], CURRENT_VCB                         ;
                                    00         B3   DD 00022    PUSHL   @CURRENT_VCB                                    ; 5602
                                          E6F8 CF        01   FB 00025    CALLS   #1, DELETE_WINDOW                       ;
                                    50              63   D0 0002A    MOVL    CURRENT_VCB, R0                         ; 5603
                                                    60   D4 0002D    CLRL    (R0)                                    ;
                                               10   A0   D5 0002F    TSTL    16(R0)                                  ; 5607
                                                    11   13 00032    BEQL    1$                                      ;
                                               10   A0   DD 00034    PUSHL   16(R0)                                  ; 5610
                                 7E       1C   A0        08   C7 00037    DIVL3   #8, 28(R0), -(SP)                       ;
                                    64              02   FB 0003C    CALLS   #2, FREE_VM                             ;
                                    50              63   D0 0003F    MOVL    CURRENT_VCB, R0                         ; 5611
                                               10   A0   D4 00042    CLRL    16(R0)                                  ;
                                    50              63   D0 00045 1$: MOVL    CURRENT_VCB, R0                         ; 5617
                                    52         28   B0   0F 00048    REMQUE  @40(R0), ACB                            ;
                                                    09   1D 0004C    BVS     2$                                      ;
                                                    52   DD 0004E    PUSHL   ACB                                     ; 5620
                                                    10   DD 00050    PUSHL   #16                                     ;
                                    64              02   FB 00052    CALLS   #2, FREE_VM                             ;
                                                    EE   11 00055    BRB     1$                                      ; 5617
                                    50              63   D0 00057 2$: MOVL    CURRENT_VCB, R0                         ; 5626
                                       07   A0        1E   8A 0005A    BICB2   #30, 7(R0)                              ; 5629
                                    50         FC   A3   D0 0005E    MOVL    CURRENT_MTL, R0                         ; 5631
                                               08   A0   DD 00062    PUSHL   8(R0)                                   ;
                                          E6B8 CF        01   FB 00065    CALLS   #1, DELETE_WINDOW                       ;
                                    50         FC   A3   D0 0006A    MOVL    CURRENT_MTL, R0                         ; 5632
                                               08   A0   D4 0006E    CLRL    8(R0)                                   ;
                                                    04 00071         RET                                             ; 5634

; Routine Size:  114 bytes,    Routine Base:  CODE + 22CA
```

```
;  4112          5635  1 %SBTTL 'READY_DISK - make save set sequential disk ready'
;  4113          5636  1 GLOBAL ROUTINE READY_DISK (MOUNT_MODE) =
;  4114          5637  1
;  4115          5638  1 !++
;  4116          5639  1 !
;  4117          5640  1 ! FUNCTIONAL DESCRIPTION:
;  4118          5641  1 !
;  4119          5642  1 !       This routine gets the disk ready as specified and returns the
;  4120          5643  1 !       address of the VCB.
;  4121          5644  1 !
;  4122          5645  1 ! CALLING SEQUENCE:
;  4123          5646  1 !       READY_DISK (MOUNT_MODE)
;  4124          5647  1 !
;  4125          5648  1 ! INPUT PARAMETERS:
;  4126          5649  1 !       MOUNT_MODE: mode disk is to be mounted in:
;  4127          5650  1 !                       0 = read
;  4128          5651  1 !                       1 = write, initialize
;  4129          5652  1 !                       3 = write, no initialize
;  4130          5653  1 !
;  4131          5654  1 ! IMPLICIT INPUTS:
;  4132          5655  1 !       RWSV_VOL_NUMBER: RVN of disk to mount
;  4133          5656  1 !
;  4134          5657  1 ! OUTPUT PARAMETERS:
;  4135          5658  1 !       NONE
;  4136          5659  1 !
;  4137          5660  1 ! IMPLICIT OUTPUTS:
;  4138          5661  1 !       NONE
;  4139          5662  1 !
;  4140          5663  1 ! ROUTINE VALUE:
;  4141          5664  1 !       VCB of mounted disk
;  4142          5665  1 !
;  4143          5666  1 ! SIDE EFFECTS:
;  4144          5667  1 !       NONE
;  4145          5668  1 !
;  4146          5669  1 !--
;  4147          5670  1
;  4148          5671  2 BEGIN
;  4149          5672  2
;  4150          5673  2 LOCAL
;  4151          5674  2         STATUS,                         ! system service status
;  4152          5675  2         IO_STATUS       : VECTOR [4,WORD], ! I/O status block
;  4153          5676  2         BUFFER          : BBLOCK [512], ! scratch I/O buffer
;  4154          5677  2         VCB             : REF BBLOCK;   ! VCB of volume being mounted
;  4155          5678  2
;  4156          5679  2 EXTERNAL ROUTINE
;  4157          5680  2         MOUNT_MESSAGE;                  ! issue mount request and get reply
;  4158          5681  2
;  4159          5682  2
;  4160          5683  2 ! Get the skeleton MTL and VCB's set up if they aren't yet.
;  4161          5684  2 !
;  4162          5685  2
;  4163          5686  2 IF .MOUNT_MODE THEN CURRENT_MTL = .OUTPUT_MTL ELSE CURRENT_MTL = .INPUT_MTL;
;  4164          5687  2 IF .CURRENT_MTL EQL 0
;  4165          5688  2 THEN STA_MOUNT (.MOUNT_MODE, 0);
;  4166          5689  2
;  4167          5690  2 ! Find the next VCB to use and set it up.
;  4168          5691  2 !
```

```
; 4169          5692  2
; 4170          5693  2 IF .RWSV_VOL_NUMBER - .CURRENT_MTL[MTL_RVN_BASE] GEQU .CURRENT_MTL[MTL_SETCOUNT]
; 4171          5694  2 THEN CURRENT_MTL[MTL_RVN_BASE] = .RWSV_VOL_NUMBER;
; 4172          5695  2
; 4173          5696  2 VCB = .CURRENT_MTL[MTL_VCB(.RWSV_VOL_NUMBER-.CURRENT_MTL[MTL_RVN_BASE])];
; 4174          5697  2 SWITCH_VOLUME (.RWSV_VOL_NUMBER);
; 4175          5698  2
; 4176          5699  2 ! Unless this is the first volume, start with operator assistance. Because
; 4177          5700  2 ! not all disks can be spun down, there is the chance that a previous
; 4178          5701  2 ! volume of the set is still in the drive.
; 4179          5702  2 !
; 4180          5703  2
; 4181          5704  2 IF .RWSV_SEG_NUMBER NEQ 0
; 4182          5705  2 THEN
; 4183          5706  3     BEGIN
; 4184          5707  3     IF .MOUNT_MODE
; 4185          5708  3     THEN MOUNT_MESSAGE (BACKUP$_READYWRITE)
; 4186          5709  3     ELSE MOUNT_MESSAGE (BACKUP$_READYREAD);
; 4187          5710  2     END;
; 4188          5711  2
; 4189          5712  2 ! Loop, checking for disk on line and write enabled if necessary,
; 4190          5713  2 ! prompting to the user until satisfied.
; 4191          5714  2 !
; 4192          5715  2
; 4193          5716  2 WHILE TRUE
; 4194          5717  2 DO
; 4195          5718  3     BEGIN
; 4196        P 5719  3     STATUS = $QIOW (CHAN = .VCB[VCB_CHAN],
; 4197        P 5720  3                     FUNC = IO$_PACKACK,
; 4198        P 5721  3                     IOSB = IO_STATUS
; 4199          5722  3                     );
; 4200          5723  3     IF .STATUS THEN STATUS = .IO_STATUS[0];
; 4201          5724  3     IF NOT .STATUS
; 4202          5725  3     AND .STATUS NEQ SS$_ILLIOFUNC
; 4203          5726  4     AND (.STATUS NEQ SS$_NOPRIV
; 4204          5727  4          OR .RWSV_SEG_NUMBER GEQU .CURRENT_MTL[MTL_SETCOUNT])
; 4205          5728  3     THEN FILE_ERROR (IF .MOUNT_MODE THEN BACKUP$_OPENOUT+STS$K_SEVERE
; 4206          5729  3                      ELSE BACKUP$_OPENIN+STS$K_SEVERE, .VCB[VCB_FAB], .STATUS);
; 4207          5730  3
; 4208        P 5731  3     STATUS = $QIOW (CHAN = .VCB[VCB_CHAN],
; 4209        P 5732  3                     FUNC = IO$_READPBLK,
; 4210        P 5733  3                     IOSB = IO_STATUS,
; 4211        P 5734  3                     P1   = BUFFER,
; 4212        P 5735  3                     P2   = 512,
; 4213        P 5736  3                     P3   = 0
; 4214          5737  3                     );
; 4215          5738  3     IF .STATUS THEN STATUS = .IO_STATUS[0];
; 4216          5739  3     IF .STATUS EQL SS$_MEDOFL
; 4217          5740  3     THEN
; 4218          5741  4         BEGIN
; 4219          5742  4         IF .MOUNT_MODE
; 4220          5743  4         THEN MOUNT_MESSAGE (BACKUP$_READYWRITE)
; 4221          5744  4         ELSE MOUNT_MESSAGE (BACKUP$_READYREAD);
; 4222          5745  4         END
; 4223          5746  4
; 4224          5747  3     ELSE IF NOT .STATUS
; 4225          5748  3     THEN FILE_ERROR (IF .MOUNT_MODE THEN BACKUP$_OPENOUT+STS$K_SEVERE
```

```
;  4226          5749  3                       ELSE BACKUP$_OPENIN+STS$K_SEVERE, .VCB[VCB_FAB], .STATUS)
;  4227          5750  3
;  4228          5751  3     ELSE IF .MOUNT_MODE
;  4229          5752  3     THEN
;  4230          5753  4         BEGIN
;  4231        P 5754  4         STATUS = $QIOW (CHAN = .VCB[VCB_CHAN],
;  4232        P 5755  4                         FUNC = IO$_WRITELBLK,
;  4233        P 5756  4                         IOSB = IO_STATUS,
;  4234        P 5757  4                         P1   = BUFFER,
;  4235        P 5758  4                         P2   = 512,
;  4236        P 5759  4                         P3   = 0
;  4237          5760  4                         );
;  4238          5761  4         IF .STATUS THEN STATUS = .IO_STATUS[0];
;  4239          5762  4         IF .STATUS EQL SS$_WRITLCK
;  4240          5763  4         THEN MOUNT_MESSAGE (BACKUP$_WRITENABLE)
;  4241          5764  4
;  4242          5765  4         ELSE IF NOT .STATUS
;  4243          5766  4         THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .VCB[VCB_FAB], .STATUS)
;  4244          5767  4
;  4245          5768  4         ELSE EXITLOOP;
;  4246          5769  4         END
;  4247          5770  4
;  4248          5771  3     ELSE EXITLOOP;
;  4249          5772  2     END;
;  4250          5773  2
;  4251          5774  2 STA_MOUNT (.MOUNT_MODE, .RWSV_VOL_NUMBER);
;  4252          5775  2
;  4253          5776  2 .VCB
;  4254          5777  1 END;                                    ! End of routine READY_DISK


                                                                .EXTRN  MOUNT_MESSAGE

                                                  03FC 00000    .ENTRY  READY_DISK, Save R2,R3,R4,R5,R6,R7,R8,R9    ; 5636
                                59 00000000G      00   9E 00002    MOVAB   FILE_ERROR, R9                          ;
                                58 00000000G      8F   D0 00009    MOVL    #BACKUP$_OPENIN+4, R8                   ;
                                57 00000000G      8F   D0 00010    MOVL    #BACKUP$_OPENOUT+4, R7                  ;
                                56 00000000G      00   9E 00017    MOVAB   SYS$QIOW, R6                            ;
                                55 00000000'      EF   9E 0001E    MOVAB   CURRENT_MTL, R5                         ;
                                5E      FDF8      CE   9E 00025    MOVAB   -520(SP), SP                            ;
                                54        04      AC   D0 0002A    MOVL    MOUNT_MODE, R4                          ; 5686
                                06                54   E9 0002E    BLBC    R4, 1$                                  ;
                                65        FC      A5   D0 00031    MOVL    OUTPUT_MTL, CURRENT_MTL                 ;
                                                  04   11 00035    BRB     2$                                      ;
                                65        F8      A5   D0 00037 1$: MOVL    INPUT_MTL, CURRENT_MTL                  ;
                                                  09   12 0003B 2$: BNEQ    3$                                      ; 5687
                                                  7E   D4 0003D    CLRL    -(SP)                                   ; 5688
                                                  54   DD 0003F    PUSHL   R4                                      ;
                              EFA2      CF        02   FB 00041    CALLS   #2, STA_MOUNT                           ;
                                53      F928      C5   3C 00046 3$: MOVZWL  RWSV_VOL_NUMBER, R3                     ; 5693
                                50                65   D0 0004B    MOVL    CURRENT_MTL, R0                         ;
                                51        30      A0   9A 0004E    MOVZBL  48(R0), R1                              ;
                          51                      53   C3 00052    SUBL3   R1, R3, R1                              ;
                51        1F    A0              08   ED 00056    CMPZV   #0, #8, 31(R0), R1                      ;
                                                  04   1A 0005C    BGTRU   4$                                      ;
                                30    A0          53   90 0005E    MOVB    R3, 48(R0)                              ; 5694
```

```
                                51        30    A0  9A 00062 4$:     MOVZBL  48(R0), R1                      ; 5696
                             51 53              51  C3 00066         SUBL3   R1, R3, R1                      ;
                                52        34 A041  D0 0006A         MOVL    52(R0)[R1], VCB                 ;
                                                53  DD 0006F         PUSHL   R3                              ; 5697
                          DD8F  CF                  01  FB 00071         CALLS   #1, SWITCH_VOLUME               ;
                                              F92A  C5  B5 00076         TSTW    RWSV_SEG_NUMBER                 ; 5704
                                                    10  13 0007A         BEQL    7$                              ;
                                                 008C  31 0007C         BRW     14$                             ; 5707
                                     00000000G  8F    DD 0007F 5$:      PUSHL   #BACKUP$_READYREAD              ; 5709
                              00000000G  00           01  FB 00085 6$:      CALLS   #1, MOUNT_MESSAGE               ;
                                                      7E  7C 0008C 7$:      CLRQ    -(SP)                           ; 5722
                                                      7E  7C 0008E         CLRQ    -(SP)                           ;
                                                      7E  7C 00090         CLRQ    -(SP)                           ;
                                                      7E  7C 00092         CLRQ    -(SP)                           ;
                                                  F8  AD  9F 00094         PUSHAB  IO_STATUS                       ;
                                                      08  DD 00097         PUSHL   #8                              ;
                                                7E        08  A2  3C 00099         MOVZWL  8(VCB), -(SP)                   ;
                                                      7E  D4 0009D         CLRL    -(SP)                           ;
                                                      66  0C  FB 0009F         CALLS   #12, SYS$QIOW                   ;
                                                      53  50  D0 000A2         MOVL    R0, STATUS                      ;
                                                      07  53  E9 000A5         BLBC    STATUS, 8$                      ; 5723
                                                      53  F8  AD  3C 000A8         MOVZWL  IO_STATUS, STATUS               ;
                                                      2D  53  E8 000AC         BLBS    STATUS, 12$                     ; 5724
                              000000F4  8F                53  D1 000AF 8$:      CMPL    STATUS, #244                    ; 5725
                                                      24  13  000B6         BEQL    12$                             ;
                                                      24  53  D1 000B8         CMPL    STATUS, #36                     ; 5726
                                                      0E  12  000BB         BNEQ    9$                              ;
                                                      50  65  D0 000BD         MOVL    CURRENT_MTL, R0                 ; 5727
                                                      51  1F  A0  9A 000C0         MOVZBL  31(R0), R1                      ;
                                                F92A  C5  51  B1 000C4         CMPW    R1, RWSV_SEG_NUMBER             ;
                                                      11  1A 000C9         BGTRU   12$                             ;
                                                      53  DD 000CB 9$:      PUSHL   STATUS                          ; 5729
                                                   30     A2  DD 000CD         PUSHL   48(VCB)                         ;
                                                04        54  E9 000D0         BLBC    R4, 10$                         ;
                                                      57  DD 000D3         PUSHL   R7                              ; 5728
                                                      02  11  000D5         BRB     11$                             ;
                                                      58  DD 000D7 10$:      PUSHL   R8                              ; 5729
                                                69        03  FB 000D9 11$:      CALLS   #3, FILE_ERROR                  ; 5728
                                                      7E  7C 000DC 12$:      CLRQ    -(SP)                           ; 5737
                                                      7E  7C 000DE         CLRQ    -(SP)                           ;
                                                7E     0200   8F  3C 000E0         MOVZWL  #512, -(SP)                     ;
                                                      14  AE  9F 000E5         PUSHAB  BUFFER                          ;
                                                      7E  7C 000E8         CLRQ    -(SP)                           ;
                                                      F8  AD  9F 000EA         PUSHAB  IO_STATUS                       ;
                                                      21  DD 000ED         PUSHL   #33                             ;
                                                7E        08  A2  3C 000EF         MOVZWL  8(VCB), -(SP)                   ;
                                                      7E  D4 000F3         CLRL    -(SP)                           ;
                                                      66  0C  FB 000F5         CALLS   #12, SYS$QIOW                   ;
                                                      53  50  D0 000F8         MOVL    R0, STATUS                      ;
                                                      04  53  E9 000FB         BLBC    STATUS, 13$                     ; 5738
                                                      53  F8  AD  3C 000FE         MOVZWL  IO_STATUS, STATUS               ;
                              000001A4  8F                53  D1 00102 13$:      CMPL    STATUS, #420                    ; 5739
                                                      0E  12  00109         BNEQ    16$                             ;
                                                      03  54  E8 0010B 14$:      BLBS    R4, 15$                         ; 5742
                                                   FF6E     31  0010E         BRW     5$                              ;
                                     00000000G  8F    DD 00111 15$.     PUSHL   #BACKUP$_READYWRITE             ; 5743
                                                      47  11  00117         BRB     19$                             ;
```

```
                                0C              53  E8 00119 16$:    BLBS    STATUS, 17$                             ; 5747
                                                53  DD 0011C        PUSHL   STATUS                                  ; 5749
                                    30          A2  DD 0011E        PUSHL   48(VCB)                                 ;
                                47              54  E8 00121        BLBS    R4, 21$                                 ;
                                                58  DD 00124        PUSHL   R8                                      ;
                                                45  11 00126        BRB     22$                                     ; 5748
                                48              54  E9 00128 17$:    BLBC    R4, 23$                                 ; 5751
                                                7E  7C 0012B        CLRQ    -(SP)                                   ; 5760
                                                7E  7C 0012D        CLRQ    -(SP)                                   ;
                                7E      0200    8F  3C 0012F        MOVZWL  #512, -(SP)                             ;
                                          14    AE  9F 00134        PUSHAB  BUFFER                                  ;
                                                7E  7C 00137        CLRQ    -(SP)                                   ;
                                          F8    AD  9F 00139        PUSHAB  IO_STATUS                               ;
                                                20  DD 0013C        PUSHL   #32                                     ;
                                7E        08    A2  3C 0013E        MOVZWL  8(VCB), -(SP)                           ;
                                                7E  D4 00142        CLRL    -(SP)                                   ;
                                66              0C  FB 00144        CALLS   #12, SYS$QIOW                           ;
                                53              50  D0 00147        MOVL    R0, STATUS                              ;
                                04              53  E9 0014A        BLBC    STATUS, 18$                             ; 5761
                                53        F8    AD  3C 0014D        MOVZWL  IO_STATUS, STATUS                       ;
                     0000025C   8F              53  D1 00151 18$:    CMPL    STATUS, #604                            ; 5762
                                                09  12 00158        BNEQ    20$                                     ;
                                    00000000G   8F  DD 0015A        PUSHL   #BACKUP$_WRITENABLE                     ; 5763
                                          FF22  31  00160 19$:      BRW     6$                                      ;
                                0D              53  E8 00163 20$:    BLBS    STATUS, 23$                             ; 5765
                                                53  DD 00166        PUSHL   STATUS                                  ; 5766
                                    30          A2  DD 00168        PUSHL   48(VCB)                                 ;
                                                57  DD 0016B 21$:    PUSHL   R7                                      ;
                                69              03  FB 0016D 22$:    CALLS   #3, FILE_ERROR                          ;
                                          FF19  31  00170        BRW     7$                                      ;
                                7E      F928    C5  3C 00173 23$:    MOVZWL  RWSV_VOL_NUMBER, -(SP)                  ; 5774
                                                54  DD 00178        PUSHL   R4                                      ;
                          EE69  CF              02  FB 0017A        CALLS   #2, STA_MOUNT                           ;
                                50              52  D0 0017F        MOVL    VCB, R0                                 ; 5777
                                                    04 00182        RET                                             ;

; Routine Size:  387 bytes,    Routine Base:  CODE + 233C
```

```
;  4256          5778  1 %SBTTL 'Directory context vector'
;  4257          5779  1 ! Entries in directory processing context vector
;  4258          5780  1 !
;  4259          5781  1
;  4260          5782  1 LITERAL
;  4261          5783  1         CTX_CHANNEL     = 0,            ! channel to use for directory
;  4262          5784  1         CTX_COUNT       = 1,            ! count of name string
;  4263          5785  1         CTX_STRING      = 2,            ! address of name string
;  4264          5786  1         CTX_VERSION     = 3,            ! version number
;  4265          5787  1         CTX_VBN         = 4,            ! VBN of directory block
;  4266          5788  1         CTX_BUFFER      = 5,            ! buffer address of current directory block
;  4267          5789  1         CTX_ENTRY       = 6,            ! address of directory record
;  4268          5790  1         CTX_FILEVER     = 7,            ! address of directory version entry
;  4269          5791  1         CTX_FAB         = 8,            ! FAB for operation
;  4270          5792  1         CTX_EOF         = 9;            ! size of directory file
```

```
;4272           5793  1 %SBTTL 'NEXT_REC - find next directory record'
;4273           5794  1 ROUTINE NEXT_REC (ENTRY, DIR_CONTEXT) =
;4274           5795  1
;4275           5796  1 !++
;4276           5797  1 !
;4277           5798  1 ! FUNCTIONAL DESCRIPTION:
;4278           5799  1 !
;4279           5800  1 !       This routine locates the next directory record and checks it for
;4280           5801  1 !       consistency.
;4281           5802  1 !
;4282           5803  1 ! CALLING SEQUENCE:
;4283           5804  1 !       NEXT_REC (ENTRY, DIR_CONTEXT)
;4284           5805  1 !
;4285           5806  1 ! INPUT PARAMETERS:
;4286           5807  1 !       ENTRY: address of present record
;4287           5808  1 !       DIR_CONTEXT: pointer to context block
;4288           5809  1 !
;4289           5810  1 ! IMPLICIT INPUTS:
;4290           5811  1 !       NONE
;4291           5812  1 !
;4292           5813  1 ! OUTPUT PARAMETERS:
;4293           5814  1 !       NONE
;4294           5815  1 !
;4295           5816  1 ! IMPLICIT OUTPUTS:
;4296           5817  1 !       NONE
;4297           5818  1 !
;4298           5819  1 ! ROUTINE VALUE:
;4299           5820  1 !       address of next directory record
;4300           5821  1 !
;4301           5822  1 ! SIDE EFFECTS:
;4302           5823  1 !       NONE
;4303           5824  1 !
;4304           5825  1 !--
;4305           5826  1
;4306           5827  2 BEGIN
;4307           5828  2
;4308           5829  2 MAP
;4309           5830  2         ENTRY           : REF BBLOCK,   ! current directory record
;4310           5831  2         DIR_CONTEXT     : REF VECTOR;   ! context block
;4311           5832  2
;4312           5833  2 LOCAL
;4313           5834  2         NEXT_ENTRY      : REF BBLOCK;   ! new directory record
;4314           5835  2
;4315           5836  2 BIND
;4316           5837  2         CHANNEL         = DIR_CONTEXT[CTX_CHANNEL],
;4317           5838  2         FND_COUNT       = DIR_CONTEXT[CTX_COUNT],
;4318           5839  2         FND_STRING      = DIR_CONTEXT[CTX_STRING],
;4319           5840  2         FND_VERSION     = DIR_CONTEXT[CTX_VERSION],
;4320           5841  2         DIR_VBN         = DIR_CONTEXT[CTX_VBN],
;4321           5842  2         DIR_BUFFER      = DIR_CONTEXT[CTX_BUFFER] : REF BBLOCK,
;4322           5843  2         DIR_ENTRY       = DIR_CONTEXT[CTX_ENTRY] : REF BBLOCK,
;4323           5844  2         DIR_VERSION     = DIR_CONTEXT[CTX_FILEVER] : REF BBLOCK,
;4324           5845  2         DIR_FAB         = DIR_CONTEXT[CTX_FAB],
;4325           5846  2         LAST_BLOCK      = DIR_CONTEXT[CTX_EOF];
;4326           5847  2
;4327           5848  2
;4328           5849  2 ! Find the next record by adding in the record size of the current entry.
```

```
;  4329          5850  2 ! The count field of the next entry must be either 65535 or be contained
;  4330          5851  2 ! within the block and even. Check the legality of the record type field.
;  4331          5852  2 !
;  4332          5853  2
;  4333          5854  2 IF .ENTRY[DIR$W_SIZE] LSSU DIR$C_LENGTH + DIR$C_VERSION
;  4334          5855  2 THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, SS$_BADIRECTORY);
;  4335          5856  2 NEXT_ENTRY = .ENTRY + .ENTRY[DIR$W_SIZE] + 2;
;  4336          5857  2 IF .NEXT_ENTRY GEQA .DIR_BUFFER + 512
;  4337          5858  2 THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, SS$_BADIRECTORY);
;  4338          5859  2
;  4339          5860  2 IF .NEXT_ENTRY[DIR$W_SIZE] NEQ 65535
;  4340          5861  2 THEN
;  4341          5862  2     IF .NEXT_ENTRY<0,1>
;  4342          5863  2     OR .(NEXT_ENTRY[DIR$W_SIZE])<0,1>
;  4343          5864  2     OR .NEXT_ENTRY[DIR$V_TYPE] NEQ DIR$C_FID
;  4344          5865  2     THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, SS$_BADIRECTORY);
;  4345          5866  2
;  4346          5867  2 RETURN .NEXT_ENTRY
;  4347          5868  2
;  4348          5869  1 END;                                    ! end of routine NEXT_REC


                                    003C 00000 NEXT_REC:
                                                        .WORD   Save R2,R3,R4,R5                        ; 5794
                                55 00000000G   8F   D0 00002        MOVL    #BACKUP$_OPENOUT+4, R5      ;
                                54 00000000G   00   9E 00009        MOVAB   FILE_ERROR, R4              ;
                                53         08   AC   D0 00010        MOVL    DIR_CONTEXT, R3             ; 5837
                                0E         04   BC   B1 00014        CMPW    @ENTRY, #14                 ; 5854
                                                0D   1E 00018        BGEQU   1$                          ;
                                           7E 0828   8F   3C 0001A        MOVZWL  #2088, -(SP)            ; 5855
                                                20   A3   DD 0001F        PUSHL   32(R3)                  ;
                                                55   DD   00022        PUSHL   R5                      ;
                                                64        03   FB 00024        CALLS   #3, FILE_ERROR      ;
                                                52   04   BC   3C 00027 1$:     MOVZWL  @ENTRY, R2          ; 5856
                                                52   04   AC   C0 0002B        ADDL2   ENTRY, R2           ;
                                                52        02   C0 0002F        ADDL2   #2, NEXT_ENTRY      ;
                                50         14   A3 00000200   8F   C1 00032        ADDL3   #512, 20(R3), R0    ; 5857
                                                50        52   D1 0003B        CMPL    NEXT_ENTRY, R0      ;
                                                0D   1F 0003E        BLSSU   2$                          ;
                                           7E 0828   8F   3C 00040        MOVZWL  #2088, -(SP)            ; 5858
                                                20   A3   DD 00045        PUSHL   32(R3)                  ;
                                                55   DD   00048        PUSHL   R5                      ;
                                                64        03   FB 0004A        CALLS   #3, FILE_ERROR      ;
                                      FFFF      8F        62   B1 0004D 2$:     CMPW    (NEXT_ENTRY), #65535    ; 5860
                                                19   13 00052        BEQL    4$                          ;
                                                09        52   E8 00054        BLBS    NEXT_ENTRY, 3$      ; 5862
                                                06        62   E8 00057        BLBS    (NEXT_ENTRY), 3$    ; 5863
                                                07   04   A2   93 0005A        BITB    4(NEXT_ENTRY), #7   ; 5864
                                                0D   13 0005E        BEQL    4$                          ;
                                           7E 0828   8F   3C 00060 3$:     MOVZWL  #2088, -(SP)            ; 5865
                                                20   A3   DD 00065        PUSHL   32(R3)                  ;
                                                55   DD   00068        PUSHL   R5                      ;
                                                64        03   FB 0006A        CALLS   #3, FILE_ERROR      ;
                                                50        52   D0 0006D 4$:     MOVL    NEXT_ENTRY, R0      ; 5867
```

```
                04 00070        RET                                     ; 5869

; Routine Size:  113 bytes,    Routine Base:  CODE + 24BF
```

```
;  4350          5870  1 %SBTTL 'DIR_SCAN - scan directory file'
;  4351          5871  1 ROUTINE DIR_SCAN (DIR_CONTEXT) =
;  4352          5872  1
;  4353          5873  1 !++
;  4354          5874  1 !
;  4355          5875  1 ! FUNCTIONAL DESCRIPTION:
;  4356          5876  1 !
;  4357          5877  1 !       This routine scans a directory, searching for the given entry.
;  4358          5878  1 !
;  4359          5879  1 !
;  4360          5880  1 ! CALLING SEQUENCE:
;  4361          5881  1 !       DIR_SCAN (DIR_CONTEXT)
;  4362          5882  1 !
;  4363          5883  1 ! INPUT PARAMETERS:
;  4364          5884  1 !       DIR_CONTEXT: context block for directory
;  4365          5885  1 !
;  4366          5886  1 ! IMPLICIT INPUTS:
;  4367          5887  1 !       NONE
;  4368          5888  1 !
;  4369          5889  1 ! OUTPUT PARAMETERS:
;  4370          5890  1 !       NONE
;  4371          5891  1 !
;  4372          5892  1 ! IMPLICIT OUTPUTS:
;  4373          5893  1 !       NONE
;  4374          5894  1 !
;  4375          5895  1 ! ROUTINE VALUE:
;  4376          5896  1 !       1 if entry found
;  4377          5897  1 !       0 if no match, in which case:
;  4378          5898  1 !               DIR_ENTRY = next record in collating sequence
;  4379          5899  1 !                         = 0 if whole directory scanned (name belongs off the end)
;  4380          5900  1 !               DIR_VERSION = next version in collating sequence if name & type matched
;  4381          5901  1 !                           = 0 if name or type did not match
;  4382          5902  1 !
;  4383          5903  1 ! SIDE EFFECTS:
;  4384          5904  1 !       directory blocks read
;  4385          5905  1 !
;  4386          5906  1 !--
;  4387          5907  1
;  4388          5908  2 BEGIN
;  4389          5909  2
;  4390          5910  2 MAP
;  4391          5911  2         DIR_CONTEXT     : REF VECTOR;   ! name descriptor block arg
;  4392          5912  2
;  4393          5913  2 LABEL
;  4394          5914  2         SEARCH_LOOP;                    ! body of search code
;  4395          5915  2
;  4396          5916  2 LOCAL
;  4397          5917  2         STATUS,                         ! routine return status
;  4398          5918  2         S_STATUS,                       ! system service status
;  4399          5919  2         IO_STATUS       : VECTOR [4,WORD], ! I/O status block
;  4400          5920  2         BLOCK,                          ! relative block number
;  4401          5921  2         ENTRY           : REF BBLOCK,   ! pointer to current directory record
;  4402          5922  2         P               : REF BBLOCK;   ! pointer to current directory version
;  4403          5923  2
;  4404          5924  2 BIND
;  4405          5925  2         CHANNEL         = DIR_CONTEXT[CTX_CHANNEL],
;  4406          5926  2         FND_COUNT       = DIR_CONTEXT[CTX_COUNT],
```

```
; 4407          5927  2          FND_STRING      = DIR_CONTEXT[CTX_STRING],
; 4408          5928  2          FND_VERSION     = DIR_CONTEXT[CTX_VERSION],
; 4409          5929  2          DIR_VBN         = DIR_CONTEXT[CTX_VBN],
; 4410          5930  2          DIR_BUFFER      = DIR_CONTEXT[CTX_BUFFER] : REF BBLOCK,
; 4411          5931  2          DIR_ENTRY       = DIR_CONTEXT[CTX_ENTRY] : REF BBLOCK,
; 4412          5932  2          DIR_VERSION     = DIR_CONTEXT[CTX_FILEVER] : REF BBLOCK,
; 4413          5933  2          DIR_FAB         = DIR_CONTEXT[CTX_FAB],
; 4414          5934  2          LAST_BLOCK      = DIR_CONTEXT[CTX_EOF];
; 4415          5935  2
; 4416          5936  2
; 4417          5937  2 ! Loop, scanning blocks of the directory until we hit EOF.
; 4418          5938  2 !
; 4419          5939  2
; 4420          5940  3 SEARCH_LOOP: BEGIN
; 4421          5941  3
; 4422          5942  3 STATUS = 0;
; 4423          5943  3 BLOCK = 1;
; 4424          5944  3 P = 0;
; 4425          5945  3 WHILE 1 DO
; 4426          5946  4     BEGIN
; 4427          5947  4
; 4428          5948  4     ENTRY = 0;
; 4429        P 5949  4     S_STATUS = $SQIOW (CHAN = .CHANNEL,
; 4430        P 5950  4                        IOSB = IO_STATUS,
; 4431        P 5951  4                        FUNC = IO$_READVBLK,
; 4432        P 5952  4                        P1   = .DIR_BUFFER,
; 4433        P 5953  4                        P2   = 512,
; 4434        P 5954  4                        P3   = .BLOCK
; 4435          5955  4                        );
; 4436          5956  4     IF .S_STATUS THEN S_STATUS = .IO_STATUS[0];
; 4437          5957  4     IF NOT .S_STATUS THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, .S_STATUS);
; 4438          5958  4     ENTRY = .DIR_BUFFER;
; 4439          5959  4
; 4440          5960  4 ! Loop, scanning the records of the directory. A record size of -1 indicates
; 4441          5961  4 ! the end of the block. We attempt to match name and type against the entry,
; 4442          5962  4 ! under control of the various name control flags.
; 4443          5963  4 !
; 4444          5964  4
; 4445          5965  4     UNTIL .ENTRY[DIR$W_SIZE] EQL 65535
; 4446          5966  4     DO
; 4447          5967  5         BEGIN
; 4448          5968  5         IF .ENTRY[DIR$W_SIZE] + .ENTRY + 2 GEQA .DIR_BUFFER + 512
; 4449          5969  5         THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, SS$_BADIRECTORY);
; 4450          5970  5
; 4451          5971  5         P = .ENTRY + DIR$C_LENGTH + .ENTRY[DIR$B_NAMECOUNT] + 1 AND NOT 1;
; 4452          5972  5         IF .P GEQA .DIR_BUFFER + 512 - DIR$C_VERSION
; 4453          5973  5         THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, SS$_BADIRECTORY);
; 4454          5974  5
; 4455          5975  6         IF (CASE CH$COMPARE (.ENTRY[DIR$B_NAMECOUNT],
; 4456          5976  6                              ENTRY[DIR$T_NAME],
; 4457          5977  6                             .FND_COUNT,
; 4458          5978  6                             .FND_STRING
; 4459          5979  6                             )
; 4460          5980  6             FROM -1 TO 1 OF
; 4461          5981  6                 SET
; 4462          5982  6
; 4463          5983  6                 [-1]:   0;      ! no match - dir entry precedes name
```

```
; 4464          5984  6
; 4465          5985  6                 [0]:    1;      ! match
; 4466          5986  6
; 4467          5987  7                 [1]:    BEGIN   ! no match - dir entry is past name
; 4468          5988  7                         P = 0;
; 4469          5989  7                         LEAVE SEARCH_LOOP;
; 4470          5990  6                         END;
; 4471          5991  6                 TES)
; 4472          5992  6
; 4473          5993  6 ! If the name and type match on a record, loop to process the versions of
; 4474          5994  6 ! the record.
; 4475          5995  6 !
; 4476          5996  6
; 4477          5997  5         THEN
; 4478          5998  6             BEGIN
; 4479          5999  6             UNTIL .P GEQA .ENTRY + .ENTRY[DIR$W_SIZE] + 2
; 4480          6000  6             DO
; 4481          6001  7                 BEGIN
; 4482          6002  7                 IF
; 4483          6003  8                     BEGIN
; 4484          6004  8
; 4485          6005  9                     IF (.FND_VERSION EQL 0
; 4486          6006  9                         AND NOT .ENTRY[DIR$V_PREVREC]
; 4487          6007  9                        )
; 4488          6008  8                     THEN 1
; 4489          6009  8
; 4490          6010  8                     ELSE IF .FND_VERSION GTR .P[DIR$W_VERSION]
; 4491          6011  8                     THEN LEAVE SEARCH_LOOP
; 4492          6012  8
; 4493          6013  8                     ELSE .FND_VERSION EQL .P[DIR$W_VERSION]
; 4494          6014  8
; 4495          6015  8                     END
; 4496          6016  8
; 4497          6017  7                 THEN
; 4498          6018  8                     BEGIN
; 4499          6019  8                     STATUS = 1;
; 4500          6020  8                     LEAVE SEARCH_LOOP;
; 4501          6021  7                     END;
; 4502          6022  7
; 4503          6023  7                 P = .P + DIR$C_VERSION;
; 4504          6024  6                 END;                        ! end of record scanning loop
; 4505          6025  6
; 4506          6026  6 ! We have gone through a directory record without finding a match.
; 4507          6027  6 ! If no continuation records are present, we can quit now.
; 4508          6028  6 !
; 4509          6029  6
; 4510          6030  6             IF NOT .ENTRY[DIR$V_NEXTREC]
; 4511          6031  6             THEN LEAVE SEARCH_LOOP;
; 4512          6032  6
; 4513          6033  5             END;                            ! end of record processing conditional
; 4514          6034  5
; 4515          6035  5         ENTRY = NEXT_REC (.ENTRY, .DIR_CONTEXT); ! get next record
; 4516          6036  4         END;                                ! end of block scanning loop
; 4517          6037  4
; 4518          6038  4     P = 0;
; 4519          6039  4     IF .BLOCK GEQU .LAST_BLOCK
; 4520          6040  4     THEN LEAVE SEARCH_LOOP;
```

```
;  4521          6041  4     BLOCK = .BLOCK + 1;
;  4522          6042  3     END;                                ! end of block loop
;  4523          6043  3
;  4524          6044  2 END;                                    ! end of block SEARCH_LOOP
;  4525          6045  2
;  4526          6046  2 ! Return the record pointers in global storage and return status.
;  4527          6047  2 !
;  4528          6048  2
;  4529          6049  2 DIR_VBN = .BLOCK;
;  4530          6050  2 DIR_ENTRY = .ENTRY;
;  4531          6051  2 DIR_VERSION = .P;
;  4532          6052  2
;  4533          6053  2 RETURN .STATUS;
;  4534          6054  2
;  4535          6055  1 END;                                    ! end of routine DIR_SCAN


                                        OFFC 00000 DIR_SCAN:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 5871
                    57 00000000G   8F   D0 00002           MOVL    #BACKUP$_OPENOUT+4, R7                        ;
                    5E             08   C2 00009           SUBL2   #8, SP                                        ;
                    55       04    AC   D0 0000C           MOVL    DIR_CONTEXT, R5                               ; 5925
                                   5B   D4 00010           CLRL    STATUS                                        ; 5942
                    59             01   D0 00012           MOVL    #1, BLOCK                                     ; 5943
                                   58   D4 00015           CLRL    P                                             ; 5944
                                   54   D4 00017 1$:       CLRL    ENTRY                                         ; 5948
                                   7E   7C 00019           CLRQ    -(SP)                                         ; 5955
                                   7E   D4 0001B           CLRL    -(SP)                                         ;
                                   59   DD 0001D           PUSHL   BLOCK                                         ;
                          7E       0200 8F   3C 0001F      MOVZWL  #512, -(SP)                                   ;
                                   14   A5   DD 00024      PUSHL   20(R5)                                        ;
                                   7E   7C 00027           CLRQ    -(SP)                                         ;
                                   20   AE   9F 00029      PUSHAB  IO_STATUS                                     ;
                                   31   DD 0002C           PUSHL   #49                                           ;
                                   65   DD 0002E           PUSHL   (R5)                                          ;
                                   7E   D4 00030           CLRL    -(SP)                                         ;
                00000000G          00   FB 00032           CALLS   #12, STA_QIOW                                 ;
                                   5A   50   DD 00039      MOVL    R0, S_STATUS                                  ;
                                   06   5A   E9 0003C      BLBC    S_STATUS, 2$                                  ; 5956
                                   5A   6E   3C 0003F      MOVZWL  IO_STATUS, S_STATUS                           ;
                                   0E   5A   E8 00042      BLBS    S_STATUS, 3$                                  ; 5957
                                        5A   DD 00045 2$:  PUSHL   S_STATUS                                      ;
                                   20   A5   DD 00047      PUSHL   32(R5)                                        ;
                                        57   DD 0004A      PUSHL   R7                                            ;
                00000000G          00   03   FB 0004C      CALLS   #3, FILE_ERROR                                ;
                                   54   14   A5   D0 00053 3$:  MOVL    20(R5), ENTRY                            ; 5958
                                   56        64   3C 00057 4$:  MOVZWL  (ENTRY), R6                            ; 5965
                          FFFF     8F        56   B1 0005A      CMPW    R6, #65535                              ;
                                             03   12 0005F      BNEQ    5$                                      ;
                                        009D 31   00061        BRW     15$                                     ;
                                   50   02   A446 9E 00064 5$:  MOVAB   2(ENTRY)[R6], R0                        ; 5968
                    53       14   A5 00000200 8F   C1 00069   ADDL3   #512, 20(R5), R3                        ;
                                   53        50   D1 00072      CMPL    R0, R3                                  ;
                                             11   1F 00075      BLSSU   6$                                      ;
```

```
                                7E      0828    8F  3C 00077            MOVZWL  #2088, -(SP)                            ; 5969
                                          20    A5  DD 0007C            PUSHL   32(R5)                                  ;
                                          57    DD  0007F               PUSHL   R7                                      ;
                    00000000G   00              03  FB 00081            CALLS   #3, FILE_ERROR                          ;
                                52        05    A4  9A 00088 6$:        MOVZBL  5(ENTRY), R2                            ; 5971
                                50      07 A244   9E 0008C              MOVAB   7(R2)[ENTRY], R0                        ;
                58              50              01  CB 00091            BICL3   #1, R0, P                               ;
                                50        F8    A3  9E 00095            MOVAB   -8(R3), R0                              ; 5972
                                50              58  D1 00099            CMPL    P, R0                                   ;
                                                11  1F 0009C            BLSSU   7$                                      ;
                                7E      0828    8F  3C 0009E            MOVZWL  #2088, -(SP)                            ; 5973
                                          20    A5  DD 000A3            PUSHL   32(R5)                                  ;
                                          57    DD  000A6               PUSHL   R7                                      ;
                    00000000G   00              03  FB 000A8            CALLS   #3, FILE_ERROR                          ;
    04  A5          00        06    A4            52  2D 000AF 7$:        CMPC5   R2, 6(ENTRY), #0, 4(R5), @8(R5)         ; 5976
                                          08    B5      000B6                                                           ;
                                                04  1A 000B8            BGTRU   8$                                      ;
                                                06  1E 000BA            BGEQU   9$                                      ;
                                                36  11 000BC            BRB     14$                                     ;
                                                58  D4 000BE 8$:        CLRL    P                                       ; 5988
                                                4C  11 000C0            BRB     16$                                     ; 5989
                                50        02 A644   9E 000C2 9$:        MOVAB   2(R6)[ENTRY], R0                        ; 5999
                                50              58  D1 000C7            CMPL    P, R0                                   ;
                                                23  1E 000CA            BGEQU   13$                                     ;
                                50        0C    A5  D0 000CC            MOVL    12(R5), R0                              ; 6005
                                                05  12 000D0            BNEQ    10$                                     ;
                                          04    A4  95 000D2            TSTB    4(ENTRY)                                ; 6006
                                                0E  18 000D5            BGEQ    11$                                     ;
        50          68              10              00  EC 000D7 10$:        CMPV    #0, #16, (P), R0                    ; 6010
                                                30  19 000DC            BLSS    16$                                     ;
        50          68              10              00  EC 000DE            CMPV    #0, #16, (P), R0                    ; 6013
                                                05  12 000E3            BNEQ    12$                                     ;
                                5B              01  D0 000E5 11$:        MOVL    #1, STATUS                          ; 6019
                                                24  11 000E8            BRB     16$                                     ; 6020
                                58              08  C0 000EA 12$:        ADDL2   #8, P                               ; 6023
                                                D3  11 000ED            BRB     9$                                      ; 5999
                    1A    04    A4              06  E1 000EF 13$:        BBC     #6, 4(ENTRY), 16$                       ; 6030
                                                30  BB 000F4 14$:        PUSHR   #^M<R4,R5>                              ; 6035
                        FE94    CF              02  FB 000F6            CALLS   #2, NEXT_REC                            ;
                                54              50  D0 000FB            MOVL    R0, ENTRY                               ;
                                                FF56 31 000FE            BRW     4$                                      ; 5965
                                                58  D4 00101 15$:        CLRL    P                                       ; 6038
                          24    A5              59  D1 00103            CMPL    BLOCK, 36(R5)                           ; 6039
                                                05  1E 00107            BGEQU   16$                                     ;
                                                59  D6 00109            INCL    BLOCK                                   ; 6041
                                                FF09 31 0010B            BRW     1$                                      ; 5945
                          10    A5              59  D0 0010E 16$:        MOVL    BLOCK, 16(R5)                           ; 6049
                          18    A5              54  D0 00112            MOVL    ENTRY, 24(R5)                           ; 6050
                          1C    A5              58  D0 00116            MOVL    P, 28(R5)                               ; 6051
                                50              5B  D0 0011A            MOVL    STATUS, R0                              ; 6053
                                                    04 0011D            RET                                             ; 6055

; Routine Size:  286 bytes,    Routine Base:  CODE + 2530
```

```
;  4537          6056  1 %SBTTL 'STA_ENTER - make directory entry'
;  4538          6057  1 GLOBAL ROUTINE STA_ENTER (FAB) : NOVALUE =
;  4539          6058  1
;  4540          6059  1 !++
;  4541          6060  1 !
;  4542          6061  1 ! FUNCTIONAL DESCRIPTION:
;  4543          6062  1 !
;  4544          6063  1 !       This routine enters the given file name in the specified directory.
;  4545          6064  1 !
;  4546          6065  1 ! CALLING SEQUENCE:
;  4547          6066  1 !       STA_ENTER (FAB)
;  4548          6067  1 !
;  4549          6068  1 ! INPUT PARAMETERS:
;  4550          6069  1 !       FAB: address of FAB for file to be entered
;  4551          6070  1 !
;  4552          6071  1 ! IMPLICIT INPUTS:
;  4553          6072  1 !       NONE
;  4554          6073  1 !
;  4555          6074  1 ! OUTPUT PARAMETERS:
;  4556          6075  1 !       NONE
;  4557          6076  1 !
;  4558          6077  1 ! IMPLICIT OUTPUTS:
;  4559          6078  1 !       DIR_RECORD: record number of new directory entry
;  4560          6079  1 !       DIR_BUFFER: buffer address of current directory block
;  4561          6080  1 !       DIR_ENTRY: address of directory record
;  4562          6081  1 !       DIR_VERSION: address of directory version entry
;  4563          6082  1 !       DIR_END: end of directory data
;  4564          6083  1 !
;  4565          6084  1 ! ROUTINE VALUE:
;  4566          6085  1 !       NONE
;  4567          6086  1 !
;  4568          6087  1 ! SIDE EFFECTS:
;  4569          6088  1 !       directory altered
;  4570          6089  1 !
;  4571          6090  1 !--
;  4572          6091  1
;  4573          6092  2 BEGIN
;  4574          6093  2 MAP
;  4575          6094  2         FAB             : REF BBLOCK;   ! FAB argument
;  4576          6095  2
;  4577          6096  2 LOCAL
;  4578          6097  2         STATUS,                         ! result of directory search
;  4579          6098  2         NAM             : REF BBLOCK,   ! address of name block
;  4580          6099  2         IO_STATUS       : VECTOR [4,WORD], ! I/O status block
;  4581          6100  2         BUFFER          : BBLOCK [512], ! I/O buffer to read directory
;  4582          6101  2         FIB             : BBLOCK [FIB$C_LENGTH], ! FIB to access directory
;  4583          6102  2         FIB_DESC        : VECTOR [2],   ! descriptor for above
;  4584          6103  2         RECATTR         : BBLOCK [FAT$C_LENGTH], ! record attributes buffer
;  4585          6104  2         FILECHAR        : BBLOCK [4],   ! file characteristics longword
;  4586          6105  2         ATT_CONTROL     : BBLOCK [20],  ! attribute control list
;  4587          6106  2         DIR_CONTEXT     : VECTOR [10],  ! directory context block
;  4588          6107  2         DIR_END         : REF BBLOCK,   ! end of directory data
;  4589          6108  2         NAME_LENGTH,                    ! length of file name, rounded even
;  4590          6109  2         NEW_SIZE,                       ! size of new directory entry
;  4591          6110  2         VERSIONS,                       ! version limit for entry
;  4592          6111  2         FAO_DESC        : VECTOR [2];   ! string descriptor for FAO
;  4593          6112  2
```

```
; 4594          6113  2 BIND
; 4595          6114  2         ATT_CONTROL0    = ATT_CONTROL : BBLOCK,
; 4596          6115  2         ATT_CONTROL1    = ATT_CONTROL+8 : BBLOCK,
; 4597          6116  2         ATT_CONTROL2    = ATT_CONTROL+16 : BBLOCK;
; 4598          6117  2
; 4599          6118  2 BIND
; 4600          6119  2         CHANNEL         = DIR_CONTEXT[CTX_CHANNEL],
; 4601          6120  2         FND_COUNT       = DIR_CONTEXT[CTX_COUNT],
; 4602          6121  2         FND_STRING      = DIR_CONTEXT[CTX_STRING],
; 4603          6122  2         FND_VERSION     = DIR_CONTEXT[CTX_VERSION],
; 4604          6123  2         DIR_VBN         = DIR_CONTEXT[CTX_VBN],
; 4605          6124  2         DIR_BUFFER      = DIR_CONTEXT[CTX_BUFFER] : REF BBLOCK,
; 4606          6125  2         DIR_ENTRY       = DIR_CONTEXT[CTX_ENTRY] : REF BBLOCK,
; 4607          6126  2         DIR_VERSION     = DIR_CONTEXT[CTX_FILEVER] : REF BBLOCK,
; 4608          6127  2         DIR_FAB         = DIR_CONTEXT[CTX_FAB],
; 4609          6128  2         LAST_BLOCK      = DIR_CONTEXT[CTX_EOF];
; 4610          6129  2
; 4611          6130  2 EXTERNAL ROUTINE
; 4612          6131  2         LIB$CVT_DTB:    ADDRESSING_MODE(GENERAL),
; 4613          6132  2                                         ! convert decimal to binary
; 4614          6133  2         INIT_NAMEBLOCK;                 ! initialize extended name block fields
; 4615          6134  2
; 4616          6135  2
; 4617          6136  2
; 4618          6137  2 ! Access the directory.
; 4619          6138  2 !
; 4620          6139  2
; 4621          6140  2 DIR_FAB = .FAB;
; 4622          6141  2 DIR_BUFFER = BUFFER;
; 4623          6142  2 CHANNEL = .FAB[FAB$L_STV];
; 4624          6143  2 NAM = .FAB[FAB$L_NAM];
; 4625          6144  2
; 4626          6145  2 CH$FILL ( 0, FIB$C_LENGTH, FIB) ;
; 4627          6146  2 FIB_DESC[0] = FIB$C_LENGTH;
; 4628          6147  2 FIB_DESC[1] = FIB;
; 4629          6148  2 FIB[FIB$L_ACCTL] = 0;
; 4630          6149  2 FIB[FIB$W_FID_NUM] = .NAM[NAM$W_DID_NUM];
; 4631          6150  2 FIB[FIB$W_FID_SEQ] = .NAM[NAM$W_DID_SEQ];
; 4632          6151  2 FIB[FIB$W_FID_RVN] = .NAM[NAM$W_DID_RVN];
; 4633          6152  2
; 4634          6153  2 ATT_CONTROL0[ATR$W_SIZE] = ATR$S_RECATTR;
; 4635          6154  2 ATT_CONTROL0[ATR$W_TYPE] = ATR$C_RECATTR;
; 4636          6155  2 ATT_CONTROL0[ATR$L_ADDR] = RECATTR;
; 4637          6156  2 ATT_CONTROL1[ATR$W_SIZE] = ATR$S_UCHAR;
; 4638          6157  2 ATT_CONTROL1[ATR$W_TYPE] = ATR$C_UCHAR;
; 4639          6158  2 ATT_CONTROL1[ATR$L_ADDR] = FILECHAR;
; 4640          6159  2 ATT_CONTROL2[0,0,32,0] = 0;
; 4641          6160  2
; 4642        P 6161  2 STATUS = $SQIOW (CHAN = .CHANNEL,
; 4643        P 6162  2                  IOSB = IO_STATUS,
; 4644        P 6163  2                  FUNC = IO$_ACCESS OR IO$M_ACCESS,
; 4645        P 6164  2                  P1   = FIB_DESC,
; 4646        P 6165  2                  P5   = ATT_CONTROL
; 4647          6166  2                  );
; 4648          6167  2 IF .STATUS THEN STATUS = .IO_STATUS[0];
; 4649          6168  2 IF NOT .STATUS THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, .STATUS);
; 4650          6169  2
```

```
; 4651          6170  2 LAST_BLOCK = ROT (.RECATTR[FAT$L_EFBLK], 16);
; 4652          6171  2 IF .RECATTR[FAT$W_FFBYTE] EQL 0
; 4653          6172  2 AND .LAST_BLOCK NEQ 0
; 4654          6173  2 THEN LAST_BLOCK = .LAST_BLOCK - 1;
; 4655          6174  2
; 4656          6175  2 IF NOT .FILECHAR[FCH$V_DIRECTORY]
; 4657          6176  2 OR .RECATTR[FAT$B_RTYPE] NEQ FAT$C_VARIABLE
; 4658          6177  2 OR .RECATTR[FAT$B_RATTRIB] NEQ FAT$M_NOSPAN
; 4659          6178  2 OR .LAST_BLOCK EQL 0
; 4660          6179  2 THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, SS$_BADIRECTORY);
; 4661          6180  2
; 4662          6181  2 ! Search the directory for the indicated name. If the search succeeds, we
; 4663          6182  2 ! have a duplicate entry. If the search failed, make a new entry.
; 4664          6183  2 !
; 4665          6184  2
; 4666          6185  2 FND_STRING = .NAM[NAM$L_NAME];
; 4667          6186  2 FND_COUNT = .NAM[NAM$L_VER] - .NAM[NAM$L_NAME];
; 4668          6187  2 IF .NAM[NAM$B_VER] LSSU 2
; 4669          6188  2 THEN FND_VERSION = 0
; 4670          6189  2 ELSE IF NOT LIB$CVT_DTB (.NAM[NAM$B_VER]-1, .NAM[NAM$L_VER]+1, FND_VERSION)
; 4671          6190  2 THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, SS$_BADFILEVER);
; 4672          6191  2 IF .FND_VERSION GTRU 32767
; 4673          6192  2 THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, SS$_BADFILEVER);
; 4674          6193  2
; 4675          6194  2 STATUS =  DIR_SCAN (DIR_CONTEXT);
; 4676          6195  2 IF .STATUS
; 4677          6196  2 AND .FND_VERSION NEQ 0
; 4678          6197  2 THEN
; 4679          6198  2     FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, SS$_DUPFILENAME);
; 4680          6199  2
; 4681          6200  2 ! Set up the position for the insert. Scan to the end of the
; 4682          6201  2 ! records in the block.
; 4683          6202  2 !
; 4684          6203  2
; 4685          6204  2 NAME_LENGTH = .FND_COUNT + 1 AND NOT 1;
; 4686          6205  2 DIR_END = .DIR_ENTRY;
; 4687          6206  2
; 4688          6207  2 UNTIL .DIR_END[DIR$W_SIZE] EQL 65535
; 4689          6208  2 DO DIR_END = NEXT_REC (.DIR_END, DIR_CONTEXT);
; 4690          6209  2 DIR_END = .DIR_END + 2;
; 4691          6210  2
; 4692          6211  2 ! If there was not a name match, we are constructing a whole new record.
; 4693          6212  2 ! Compute the record size and see if there is enough space. If not, extend
; 4694          6213  2 ! the directory. Then shuffle down the rest of the records and build the
; 4695          6214  2 ! new entry.
; 4696          6215  2 !
; 4697          6216  2
; 4698          6217  2 IF .DIR_VERSION EQL 0
; 4699          6218  2 THEN
; 4700          6219  3     BEGIN
; 4701          6220  3     IF .FND_VERSION EQL 0
; 4702          6221  3     THEN FND_VERSION = .FND_VERSION + 1;
; 4703          6222  3
; 4704          6223  3     NEW_SIZE = DIR$C_LENGTH + DIR$C_VERSION + .NAME_LENGTH;
; 4705          6224  3     IF .NEW_SIZE GTRU .DIR_BUFFER + 512 - .DIR_END
; 4706          6225  3     THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, SS$_DIRFULL);
; 4707          6226  3
```

```
; 4708          6227  3     CH$MOVE (.DIR_END-.DIR_ENTRY, .DIR_ENTRY, .DIR_ENTRY+.NEW_SIZE);
; 4709          6228  3
; 4710          6229  3     DIR_ENTRY[DIR$W_SIZE] = .NEW_SIZE - 2;
; 4711          6230  3     VERSIONS = (512*DIR$C_VERSION-2-.NEW_SIZE) / DIR$C_VERSION;
; 4712          6231  3     IF .RECATTR[FAT$W_VERSIONS] NEQ 0
; 4713          6232  3     THEN VERSIONS = .RECATTR[FAT$W_VERSIONS];
; 4714          6233  3     DIR_ENTRY[DIR$W_VERLIMIT] = .VERSIONS;
; 4715          6234  3
; 4716          6235  3     DIR_ENTRY[DIR$B_FLAGS] = DIR$C_FID;
; 4717          6236  3     DIR_ENTRY[DIR$B_NAMECOUNT] = .FND_COUNT;
; 4718          6237  3     CH$COPY (.FND_COUNT, .FND_STRING,
; 4719          6238  3                 0, .NAME_LENGTH, DIR_ENTRY[DIR$T_NAME]);
; 4720          6239  3
; 4721          6240  3     DIR_VERSION = .DIR_ENTRY + .NEW_SIZE - DIR$C_VERSION;
; 4722          6241  3     END
; 4723          6242  3
; 4724          6243  3
; 4725          6244  3 ! Otherwise we are adding a new version to an existing entry. Then
; 4726          6245  3 ! shuffle the rest of the directory down.
; 4727          6246  3 !
; 4728          6247  3
; 4729          6248  2 ELSE
; 4730          6249  3     BEGIN
; 4731          6250  3     IF   .FND_VERSION EQL 0
; 4732          6251  3     THEN FND_VERSION = .DIR_VERSION[DIR$W_VERSION] + 1;
; 4733          6252  3
; 4734          6253  3     IF DIR$C_VERSION GTRU .DIR_BUFFER + 512 - .DIR_END
; 4735          6254  3     THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, SS$_DIRFULL);
; 4736          6255  3     DIR_ENTRY[DIR$W_SIZE] = .DIR_ENTRY[DIR$W_SIZE] + DIR$C_VERSION;
; 4737          6256  3
; 4738          6257  3     CH$MOVE (.DIR_END-.DIR_VERSION, .DIR_VERSION, .DIR_VERSION+DIR$C_VERSION);
; 4739          6258  2     END;
; 4740          6259  2
; 4741          6260  2 ! Now insert the version number and file ID into the version slot.
; 4742          6261  2 !
; 4743          6262  2
; 4744          6263  2 IF .FND_VERSION GTRU 32767
; 4745          6264  2 THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, SS$_BADFILEVER);
; 4746          6265  2 DIR_VERSION[DIR$W_VERSION] = .FND_VERSION;
; 4747          6266  2 DIR_VERSION[DIR$W_FID_NUM] = .NAM[NAM$W_FID_NUM];
; 4748          6267  2 DIR_VERSION[DIR$W_FID_SEQ] = .NAM[NAM$W_FID_SEQ];
; 4749          6268  2 DIR_VERSION[DIR$B_FID_NMX] = .NAM[NAM$B_FID_NMX];
; 4750          6269  2 DIR_VERSION[DIR$B_FID_RVN] = 0;
; 4751          6270  2
; 4752        P 6271  2 STATUS = $$QIOW (CHAN = .CHANNEL,
; 4753        P 6272  2                  IOSB = IO_STATUS,
; 4754        P 6273  2                  FUNC = IO$_WRITEVBLK,
; 4755        P 6274  2                  P1   = .DIR_BUFFER,
; 4756        P 6275  2                  P2   = 512,
; 4757        P 6276  2                  P3   = .DIR_VBN
; 4758          6277  2                  );
; 4759          6278  2 IF .STATUS THEN STATUS = .IO_STATUS[0];
; 4760          6279  2 IF NOT .STATUS THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, .STATUS);
; 4761          6280  2
; 4762        P 6281  2 STATUS = $$QIOW (CHAN = .CHANNEL,
; 4763        P 6282  2                  IOSB = IO_STATUS,
; 4764        P 6283  2                  FUNC = IO$_DEACCESS
```

```
;  4765          6284  2                    );
;  4766          6285  2 IF .STATUS THEN STATUS = .IO_STATUS[0];
;  4767          6286  2 IF NOT .STATUS THEN FILE_ERROR (BACKUP$_OPENOUT+STS$K_SEVERE, .DIR_FAB, .STATUS);
;  4768          6287  2
;  4769          6288  2 ! Finally build the resultant string with version number.
;  4770          6289  2 !
;  4771          6290  2
;  4772          6291  2 IF .NAM[NAM$B_RSL] EQL 0
;  4773          6292  2 THEN
;  4774          6293  3     BEGIN
;  4775          6294  3     CH$MOVE (.NAM[NAM$B_ESL], .NAM[NAM$L_ESA], .NAM[NAM$L_RSA]);
;  4776          6295  3     FAO_DESC[1] = .NAM[NAM$L_VER] + 1 + .NAM[NAM$L_RSA] - .NAM[NAM$L_ESA];
;  4777          6296  3     FAO_DESC[0] = .NAM[NAM$B_RSS] - (.FAO_DESC[1] - .NAM[NAM$L_RSA]);
;  4778          6297  3     IF .FAO_DESC[0] GTR 0
;  4779          6298  3     THEN
;  4780          6299  4         BEGIN
;  4781        P 6300  4         $FAO ($DESCRIPTOR ('!UW'),
;  4782        P 6301  4               FAO_DESC[0],
;  4783        P 6302  4               FAO_DESC[0],
;  4784        P 6303  4               .FND_VERSION
;  4785          6304  4               );
;  4786          6305  4         END
;  4787          6306  3     ELSE
;  4788          6307  3         FAO_DESC[0] = 0;
;  4789          6308  3     NAM[NAM$B_RSL] = .FAO_DESC[1] + .FAO_DESC[0] - .NAM[NAM$L_RSA];
;  4790          6309  3     INIT_NAMEBLOCK (.NAM);
;  4791          6310  2     END;
;  4792          6311  2
;  4793          6312  1 END;                                    ! end of routine STA_ENTER


                                57  55  21   0264E P.AAT:  .ASCII  \!UW\                                    ;
                                             02651        .BLKB   3
                                  00000003   02654 P.AAS:  .LONG   3                                         ;
                                  00000000'  02658        .ADDRESS P.AAT                                    ;

                                                          .EXTRN  LIB$CVT_DTB, INIT_NAMEBLOCK

                                                 0FFC 00000        .ENTRY  STA_ENTER, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 6057
                                                                           R10,R11
                                      5E    FD48   CE  9E 00002          MOVAB   -696(SP), SP                       ;
                                      50      04   AC  D0 00007          MOVL    FAB, R0                            ; 6140
                                2C    AE           50  D0 0000B          MOVL    R0, DIR_FAB                        ;
                                20    AE    00B0   CE  9E 0000F          MOVAB   BUFFER, DIR_BUFFER                 ; 6141
                                0C    AE      0C   A0  D0 00015          MOVL    12(R0), CHANNEL                    ; 6142
                                      56      28   A0  D0 0001A          MOVL    40(R0), NAM                        ; 6143
 0040   8F          00                 6E           00  2C 0001E          MOVC5   #0, (SP), #0, #64, FIB             ; 6145
                                      70    AE         00025                                                      ;
                                68    AE      40   8F  9A 00027          MOVZBL  #64, FIB_DESC                      ; 6146
                                6C    AE      70   AE  9E 0002C          MOVAB   FIB, FIB_DESC+4                    ; 6147
                                      70    AE         D4 00031          CLRL    FIB                                ; 6148
                                74    AE      2A   A6  D0 00034          MOVL    42(NAM), FIB+4                     ; 6149
                                78    AE      2E   A6  B0 00039          MOVW    46(NAM), FIB+8                     ; 6151
                                34    AE  00040020   8F  D0 0003E          MOVL    #262176, ATT_CONTROL0              ; 6153
                                38    AE      48   AE  9E 00046          MOVAB   RECATTR, ATT_CONTROL0+4            ; 6155
                                3C    AE  00030004   8F  D0 0004B          MOVL    #196612, ATT_CONTROL1              ; 6156
```

```
                        40   AE                  6E  9E 00053          MOVAB   FILECHAR, ATT_CONTROL1+4                    ; 6158
                                             44 AE  D4 00057          CLRL    ATT_CONTROL2                                ; 6159
                                             7E  D4 0005A             CLRL    -(SP)                                       ; 6166
                                             38 AE  9F 0005C          PUSHAB  ATT_CONTROL
                                             7E  7C 0005F             CLRQ    -(SP)
                                             7E  D4 00061             CLRL    -(SP)
                                             7C AE  9F 00063          PUSHAB  FIB_DESC
                                             7E  7C 00066             CLRQ    -(SP)
                                             F8 AD  9F 00068          PUSHAB  IO_STATUS
                                          7E 72  8F  9A 0006B         MOVZBL  #1T4, -(SP)
                                             34 AE  DD 0006F          PUSHL   CHANNEL
                                             7E  D4 00072             CLRL    -(SP)
                     00000000G   00          0C  FB 00074             CALLS   #12, STA_QIOW
                                 5B          50  D0 0007B             MOVL    R0, STATUS
                                 07          5B  E9 0007E             BLBC    STATUS, 1$                                  ; 6167
                                 5B      F8  AD  3C 00081             MOVZWL  IO_STATUS, STATUS
                                 12          5B  EB 00085             BLBS    STATUS, 2$                                  ; 6168
                                             5B  DD 00088 1$:         PUSHL   STATUS
                                         30  AE  DD 0008A             PUSHL   DIR_FAB
                                 00000000G   8F  DD 0008D             PUSHL   #BACKUP$_OPENOUT+4
                     00000000G   00          03  FB 00093             CALLS   #3, FILE_ERROR
                30   AE     50  AE                  10  9C 0009A 2$:  ROTL    #16, RECATTR+8, LAST_BLOCK                  ; 6170
                                             54 AE  B5 000A0          TSTW    RECATTR+12                                  ; 6171
                                             08  12 000A3             BNEQ    3$
                                             30 AE  D5 000A5          TSTL    LAST_BLOCK                                  ; 6172
                                             03  13 000A8             BEQL    3$
                                             30 AE  D7 000AA          DECL    LAST_BLOCK                                  ; 6173
                    11       01  AE                  05  E1 000AD 3$:  BBC     #5, FILECHAR+1, 4$                          ; 6175
                                 02          48 AE  91 000B2          CMPB    RECATTR, #2                                 ; 6176
                                             0B  12 000B6             BNEQ    4$
                                 08      49  AE  91 000B8             CMPB    RECATTR+1, #8                               ; 6177
                                             05  12 000BC             BNEQ    4$
                                             30 AE  D5 000BE          TSTL    LAST_BLOCK                                  ; 6178
                                             15  12 000C1             BNEQ    5$
                                          7E     0828  8F  3C 000C3 4$:  MOVZWL  #2088, -(SP)                             ; 6179
                                             30 AE  DD 000C8          PUSHL   DIR_FAB
                                 00000000G   8F  DD 000CB             PUSHL   #BACKUP$_OPENOUT+4
                     00000000G   00          03  FB 000D1             CALLS   #3, FILE_ERROR
                           14    AE      4C  A6  D0 000D8 5$:  MOVL    76(NAM), FND_STRING                                 ; 6185
                30   AE    54    A6      4C  A6  C3 000DD             SUBL3   76(NAM), 84(NAM), FND_COUNT                 ; 6186
                                 02      3D  A6  91 000E4             CMPB    61(NAM), #2                                 ; 6187
                                             05  1E 000E8             BGEQU   6$
                                             18 AE  D4 000EA          CLRL    FND_VERSION                                 ; 6188
                                             2D  11 000ED             BRB     7$
                                             18 AE  9F 000EF 6$:      PUSHAB  FND_VERSION                                 ; 6189
                    7E       54  A6                  01  C1 000F2     ADDL3   #1, 84(NAM), -(SP)
                                 7E      3D  A6  9A 000F7             MOVZBL  61(NAM), -(SP)
                                             6E  D7 000FB             DECL    (SP)
                     00000000G   00          03  FB 000FD             CALLS   #3, LIB$CVT_DTB
                                 15          50  E8 00104             BLBS    R0, 7$
                                          7E     0820  8F  3C 00107    MOVZWL  #2080, -(SP)                                ; 6190
                                             30 AE  DD 0010C          PUSHL   DIR_FAB
                                 00000000G   8F  DD 0010F             PUSHL   #BACKUP$_OPENOUT+4
                     00000000G   00          03  FB 00115             CALLS   #3, FILE_ERROR
                     00007FFF    8F      18  AE  D1 0011C 7$:  CMPL    FND_VERSION, #32767                                 ; 6191
                                             15  1B 00124             BLEQU   8$
                                          7E     0820  8F  3C 00126    MOVZWL  #2080, -(SP)                                ; 6192
```

```
                                30   AE   DD 0012B          PUSHL   DIR_FAB                                          ;
                        00000000G    8F   DD 0012E          PUSHL   #BACKUP$_OPENOUT+4                               ;
                 00000000G      00       03   FB 00134          CALLS   #3, FILE_ERROR                              ;
                                0C   AE   9F 0013B 8$:       PUSHAB  DIR_CONTEXT                                   ; 6194
                     FD91       CF       01   FB 0013E          CALLS   #1, DIR_SCAN                                ;
                                5B       50   D0 00143          MOVL    R0, STATUS                                  ;
                                1A       5B   E9 00146          BLBC    STATUS, 9$                                  ; 6195
                                18   AE   D5 00149          TSTL    FND_VERSION                                   ; 6196
                                         15   13 0014C          BEQL    9$                                          ;
                                7E  0868    8F   3C 0014E          MOVZWL  #2152, -(SP)                            ; 6198
                                30   AE   DD 00153          PUSHL   DIR_FAB                                       ;
                        00000000G    8F   DD 00156          PUSHL   #BACKUP$_OPENOUT+4                            ;
                 00000000G      00       03   FB 0015C          CALLS   #3, FILE_ERROR                              ;
           50         10   AE       01   C1 00163 9$:       ADDL3   #1, FND_COUNT, R0                             ; 6204
           58              50       01   CB 00168          BICL3   #1, R0, NAME_LENGTH                           ;
                           5A   24   AE   D0 0016C          MOVL    DIR_ENTRY, DIR_END                            ; 6205
                     FFFF  8F       6A   B1 00170 10$:      CMPW    (DIR_END), #65535                             ; 6207
                                         0F   13 00175          BEQL    11$                                         ;
                                0C   AE   9F 00177          PUSHAB  DIR_CONTEXT                                   ; 6208
                                5A   DD   0017A          PUSHL   DIR_END                                       ;
                     FCE2       CF       02   FB 0017C          CALLS   #2, NEXT_REC                                ;
                                5A       50   D0 00181          MOVL    R0, DIR_END                                 ;
                                         EA   11 00184          BRB     10$                                         ;
                                5A       02   C0 00186 11$:      ADDL2   #2, DIR_END                             ; 6209
                                28   AE   D5 00189          TSTL    DIR_VERSION                                   ; 6217
                                         72   12 0018C          BNEQ    15$                                         ;
                                18   AE   D5 0018E          TSTL    FND_VERSION                                   ; 6220
                                         03   12 00191          BNEQ    12$                                         ;
                                18   AE   D6 00193          INCL    FND_VERSION                                   ; 6221
                                57   0E   A8 9E 00196 12$:      MOVAB   14(R8), NEW_SIZE                        ; 6223
           50         20   AE       5A   C3 0019A          SUBL3   DIR_END, DIR_BUFFER, R0                       ; 6224
                           50  0200    C0   9E 0019F          MOVAB   512(R0), R0                                   ;
                           50       57   D1 001A4          CMPL    NEW_SIZE, R0                                  ;
                                    15   1B 001A7          BLEQU   13$                                           ;
                           7E  0860    8F   3C 001A9          MOVZWL  #2144, -(SP)                                  ; 6225
                                30   AE   DD 001AE          PUSHL   DIR_FAB                                       ;
                        00000000G    8F   DD 001B1          PUSHL   #BACKUP$_OPENOUT+4                            ;
                 00000000G      00       03   FB 001B7          CALLS   #3, FILE_ERROR                              ;
                           59   24   AE   D0 001BE 13$:       MOVL    DIR_ENTRY, R9                                 ; 6227
              50           5A       59   C3 001C2          SUBL3   R9, DIR_END, R0                               ;
            6749           69       50   28 001C6          MOVC3   R0, (R9), (NEW_SIZE)[R9]                      ;
              69           57       02   A3 001CB          SUBW3   #2, NEW_SIZE, (R9)                            ; 6229
                           50  FDFA    C7   9E 001CF          MOVAB   -518(R7), R0                                  ; 6230
                           50       08   C6 001D4          DIVL2   #8, R0                                        ;
                           50       50   CE 001D7          MNEGL   R0, VERSIONS                                  ;
                                66   AE   B5 001DA          TSTW    RECATTR+30                                    ; 6231
                                         04   13 001DD          BEQL    14$                                         ;
                           50   66   AE   3C 001DF          MOVZWL  RECATTR+30, VERSIONS                          ; 6232
                     02    A9       50   B0 001E3 14$:       MOVW    VERSIONS, 2(R9)                               ; 6233
                                04   A9   94 001E7          CLRB    4(R9)                                         ; 6235
                     05    A9       10   AE   90 001EA          MOVB    FND_COUNT, 5(R9)                              ; 6236
     58        00    14    BE       10   AE   2C 001EF          MOVC5   FND_COUNT, @FND_STRING, #0, NAME_LENGTH, -    ; 6238
                                06   A9        001F6                  6(R9)                                         ;
                     28    AE       F8 A749   9E 001F8          MOVAB   -8(NEW_SIZE)[R9], DIR_VERSION               ; 6240
                                         42   11 001FE          BRB     18$                                         ; 6217
                                18   AE   D5 00200 15$:       TSTL    FND_VERSION                                   ; 6250
                                         08   12 00203          BNEQ    16$                                         ;
```

```
                          18   AE      28   BE   32 00205          CVTWL   @DIR_VERSION, FND_VERSION                ; 6251
                               18   AE      28   D6 0020A          INCL    FND_VERSION                              ;
                     50        20   AE          5A   C3 0020D 16$:  SUBL3   DIR_END, DIR_BUFFER, R0                 ; 6253
                                    50    0200   C0   9E 00212          MOVAB   512(R0), R0                      ;
                                    08           50   D1 00217          CMPL    R0, #8                           ;
                                                 15   1E 0021A          BGEQU   17$                              ;
                                    7E    0860   8F   3C 0021C          MOVZWL  #2144, -(SP)                     ; 6254
                                          30     AE   DD 00221          PUSHL   DIR_FAB                          ;
                                    00000000G    8F   DD 00224          PUSHL   #BACKUP$_OPENOUT+4               ;
                          00000000G   00         03   FB 00227          CALLS   #3, FILE_ERROR                   ;
                                24    BE         08   A0 00231 17$:     ADDW2   #8, @DIR_ENTRY                   ; 6255
                                    51    28     AE   D0 00235          MOVL    DIR_VERSION, R1                  ; 6257
                     50                 5A           51   C3 00239          SUBL3   R1, DIR_END, R0                  ;
                08   A1                 61           50   28 0023D          MOVC3   R0, (R1), 8(R1)                  ;
                          00007FFF  8F         18     AE   D1 00242 18$:     CMPL    FND_VERSION, #32767              ; 6263
                                                      15   1B 0024A          BLEQU   19$                              ;
                                    7E    0820   8F   3C 0024C          MOVZWL  #2080, -(SP)                     ; 6264
                                          30     AE   DD 00251          PUSHL   DIR_FAB                          ;
                                    00000000G    8F   DD 00254          PUSHL   #BACKUP$_OPENOUT+4               ;
                          00000000G   00         03   FB 00257          CALLS   #3, FILE_ERROR                   ;
                                    50    28     AE   D0 00261 19$:     MOVL    DIR_VERSION, R0                  ; 6265
                                    60    18     AE   B0 00265          MOVW    FND_VERSION, (R0)                ;
                                02  A0    24     A6   D0 00269          MOVL    36(NAM), 2(R0)                   ; 6266
                                07  A0    29     A6   90 0026E          MOVB    41(NAM), 7(R0)                   ; 6268
                                          06     A0   94 00273          CLRB    6(R0)                            ; 6269
                                                 7E   7C 00276          CLRQ    -(SP)                            ; 6277
                                                 7E   D4 00278          CLRL    -(SP)                            ;
                                          28     AE   DD 0027A          PUSHL   DIR_VBN                          ;
                                    7E    0200   8F   3C 0027D          MOVZWL  #512, -(SP)                      ;
                                          34     AE   DD 00282          PUSHL   DIR_BUFFER                       ;
                                                 7E   7C 00285          CLRQ    -(SP)                            ;
                                          F8     AD   9F 00287          PUSHAB  IO_STATUS                        ;
                                          30     DD   0028A             PUSHL   #48                              ;
                                          34     AE   DD 0028C          PUSHL   CHANNEL                          ;
                                                 7E   D4 0028F          CLRL    -(SP)                            ;
                          00000000G   00         0C   FB 00291          CALLS   #12, STA_QIOW                    ;
                                    5B           50   D0 00298          MOVL    R0, STATUS                       ;
                                    07           5B   E9 0029B          BLBC    STATUS, 20$                      ; 6278
                                    5B    F8     AD   3C 0029E          MOVZWL  IO_STATUS, STATUS                ;
                                    12           5B   E8 002A2          BLBS    STATUS, 21$                      ; 6279
                                                 5B   DD 002A5 20$:     PUSHL   STATUS                           ;
                                          30     AE   DD 002A7          PUSHL   DIR_FAB                          ;
                                    00000000G    8F   DD 002AA          PUSHL   #BACKUP$_OPENOUT+4               ;
                          00000000G   00         03   FB 002B0          CALLS   #3, FILE_ERROR                   ;
                                                 7E   7C 002B7 21$:     CLRQ    -(SP)                            ; 6284
                                                 7E   7C 002B9          CLRQ    -(SP)                            ;
                                                 7E   7C 002BB          CLRQ    -(SP)                            ;
                                                 7E   7C 002BD          CLRQ    -(SP)                            ;
                                          F8     AD   9F 002BF          PUSHAB  IO_STATUS                        ;
                                          34     DD   002C2             PUSHL   #52                              ;
                                          34     AE   DD 002C4          PUSHL   CHANNEL                          ;
                                                 7E   D4 002C7          CLRL    -(SP)                            ;
                          00000000G   00         0C   FB 002C9          CALLS   #12, STA_QIOW                    ;
                                    5B           50   D0 002D0          MOVL    R0, STATUS                       ;
                                    07           5B   E9 002D3          BLBC    STATUS, 22$                      ; 6285
                                    5B    F8     AD   3C 002D6          MOVZWL  IO_STATUS, STATUS                ;
                                    12           5B   E8 002DA          BLBS    STATUS, 23$                      ; 6286
```

```
                                                    5B  DD 002DD 22$:    PUSHL   STATUS                                  ;
                                                30  AE  DD 002DF         PUSHL   DIR_FAB                                 ;
                                            00000000G  8F  DD 002E2         PUSHL   #BACKUP$_OPENOUT+4                      ;
                                    00000000G   00     03  FB 002E8         CALLS   #3, FILE_ERROR                          ;
                                                   03  A6  95 002EF 23$:    TSTB    3(NAM)                                  ; 6291
                                                       58  12 002F2         BNEQ    26$                                     ;
                                                50     0B  A6  9A 002F4         MOVZBL  11(NAM), R0                             ; 6294
                                04  B6          0C     B6      50  28 002F8         MOVC3   R0, @12(NAM), @4(NAM)           ;
                                    50          54     A6  04  A6  C1 002FE         ADDL3   4(NAM), 84(NAM), R0             ; 6295
                                                50     0C  A6  C2 00304         SUBL2   12(NAM), R0                             ;
                                                08     AE  01  A0 9E 00308         MOVAB   1(R0), FAO_DESC+4               ;
                                    50          04     A6  08  AE  C3 0030D         SUBL3   FAO_DESC+4, 4(NAM), R0          ; 6296
                                                51     02  A6  9A 00313         MOVZBL  2(NAM), R1                              ;
                                04  AE              50      51  C1 00317         ADDL3   R1, R0, FAO_DESC                ;
                                                           16  15 0031C         BLEQ    24$                                     ; 6297
                                                   18  AE  DD 0031E         PUSHL   FND_VERSION                             ; 6304
                                                   08  AE  9F 00321         PUSHAB  FAO_DESC                                ;
                                                   0C  AE  9F 00324         PUSHAB  FAO_DESC                                ;
                                                FCCD   CF  9F 00327         PUSHAB  P.AAS                                   ;
                                    00000000G   00     04  FB 0032B         CALLS   #4, SYS$FAO                             ;
                                                           03  11 00332         BRB     25$                                     ; 6297
                                                   04  AE  D4 00334 24$:    CLRL    FAO_DESC                                ; 6307
                                    50          08  AE  04  AE  C1 00337 25$:    ADDL3   FAO_DESC, FAO_DESC+4, R0        ; 6308
                                03  A6              50      04  A6  83 0033D         SUBB3   4(NAM), R0, 3(NAM)              ;
                                                           56  DD 00343         PUSHL   NAM                                     ; 6309
                                    00000000G   00         01  FB 00345         CALLS   #1, INIT_NAMEBLOCK                      ;
                                                               04 0034C 26$:    RET                                     ; 6312

; Routine Size:  845 bytes,    Routine Base:  CODE + 265C
```

```
;  4795          6313  1 %SBTTL 'STA_EXTEND - extend a file'
;  4796          6314  1 GLOBAL ROUTINE STA_EXTEND (P_COUNT, RET_COUNT) =
;  4797          6315  1
;  4798          6316  1 !++
;  4799          6317  1 !
;  4800          6318  1 ! FUNCTIONAL DESCRIPTION:
;  4801          6319  1 !
;  4802          6320  1 !       This routine attempts to allocate the specified number of
;  4803          6321  1 !       blocks to the currently open file and appends them to the file.
;  4804          6322  1 !       When asked to return blocks, allocated blocks listed in the
;  4805          6323  1 !       window are recorded in the file header.
;  4806          6324  1 !
;  4807          6325  1 ! CALLING SEQUENCE:
;  4808          6326  1 !       STA_EXTEND (P_COUNT, RET_COUNT)
;  4809          6327  1 !
;  4810          6328  1 ! INPUT PARAMETERS:
;  4811          6329  1 !       P_COUNT: if positive, the number of new blocks to allocate
;  4812          6330  1 !                if negative, the number of blocks to return; also
;  4813          6331  1 !                the signal to build header map pointers.
;  4814          6332  1 !
;  4815          6333  1 ! IMPLICIT INPUTS:
;  4816          6334  1 !       CURRENT_MTL: MTL of file in process
;  4817          6335  1 !
;  4818          6336  1 ! OUTPUT PARAMETERS:
;  4819          6337  1 !       RET_COUNT: address into which to return count of blocks allocated
;  4820          6338  1 !
;  4821          6339  1 ! IMPLICIT OUTPUTS:
;  4822          6340  1 !       NONE
;  4823          6341  1 !
;  4824          6342  1 ! ROUTINE VALUE:
;  4825          6343  1 !       success or failure status
;  4826          6344  1 !
;  4827          6345  1 ! SIDE EFFECTS:
;  4828          6346  1 !       NONE
;  4829          6347  1 !
;  4830          6348  1 !--
;  4831          6349  1
;  4832          6350  2 BEGIN
;  4833          6351  2
;  4834          6352  2 LOCAL
;  4835          6353  2         STATUS,                         ! general status value
;  4836          6354  2         BLOCK_COUNT,                    ! rounded form of requested count
;  4837          6355  2         NEW_COUNT,                      ! count of new blocks allocated
;  4838          6356  2         NEW_LBN,                        ! LBN of same
;  4839          6357  2         MTL             : REF BBLOCK,   ! address of MTL for volume set
;  4840          6358  2         W               : REF BBLOCK,   ! pointer to current window segment
;  4841          6359  2         LAST_W          : REF BBLOCK,   ! pointer to previous window
;  4842          6360  2         WP              : REF BBLOCK;   ! pointer to current window pointer
;  4843          6361  2
;  4844          6362  2
;  4845          6363  2 !  Round the block count and do setup.
;  4846          6364  2 !
;  4847          6365  2
;  4848          6366  3 BLOCK_COUNT = (.P_COUNT + .CURRENT_VCB[VCB_CLUSTER] - 1)
;  4849          6367  2                 / .CURRENT_VCB[VCB_CLUSTER] * .CURRENT_VCB[VCB_CLUSTER];
;  4850          6368  2
;  4851          6369  2 .RET_COUNT = 0;
```

```
; 4852          6370  2 MTL = .CURRENT_MTL;
; 4853          6371  2 IF .MTL[MTL_WINDOW] EQL 0 THEN RETURN SS$_FILNOTACC;
; 4854          6372  2
; 4855          6373  2 ! Attempt allocation if requested. Record the new blocks in the
; 4856          6374  2 ! window.
; 4857          6375  2 !
; 4858          6376  2
; 4859          6377  2 IF .BLOCK_COUNT GTR 0
; 4860          6378  2 THEN
; 4861          6379  3     BEGIN
; 4862          6380  3     IF NOT STA_ALLOC_BEST (.BLOCK_COUNT, NEW_COUNT, NEW_LBN)
; 4863          6381  3     THEN RETURN SS$_DEVICEFULL;
; 4864          6382  3     ADD_WINDOW_MAP (.MTL[MTL_WINDOW], .MTL[MTL_FID_RVN], .NEW_COUNT, .NEW_LBN);
; 4865          6383  3     .RET_COUNT = .NEW_COUNT;
; 4866          6384  3     END
; 4867          6385  3
; 4868          6386  3 ! Otherwise this is a request to truncate allocated blocks and finish
; 4869          6387  3 ! off the file header. First truncate the requested number of blocks
; 4870          6388  3 ! out of the window.
; 4871          6389  3 !
; 4872          6390  3
; 4873          6391  2 ELSE
; 4874          6392  3     BEGIN
; 4875          6393  3     BLOCK_COUNT = -.BLOCK_COUNT;
; 4876          6394  3     UNTIL .BLOCK_COUNT LEQ 0
; 4877          6395  3     DO
; 4878          6396  4         BEGIN
; 4879          6397  4         LAST_W = 0;
; 4880          6398  4         W = .MTL[MTL_WINDOW];
; 4881          6399  4         UNTIL .W[WCB_LINK] EQL 0
; 4882          6400  4         DO
; 4883          6401  5             BEGIN
; 4884          6402  5             LAST_W = .W;
; 4885          6403  5             W = .W[WCB_LINK];
; 4886          6404  4             END;
; 4887          6405  4         IF .W[WCB_SIZE] EQL 0 THEN EXITLOOP;
; 4888          6406  4
; 4889          6407  4         WP = .W + WCB_S_HEADER + (.W[WCB_SIZE]-1) * WCB_S_ENTRY;
; 4890          6408  4         IF .WP[WCB_COUNT] GTR .BLOCK_COUNT
; 4891          6409  4         THEN
; 4892          6410  5             BEGIN
; 4893          6411  5             WP[WCB_COUNT] = .WP[WCB_COUNT] - .BLOCK_COUNT;
; 4894          6412  5             FREE_BLOCKS (.BLOCK_COUNT, .WP[WCB_LBN] + .WP[WCB_COUNT]);
; 4895          6413  5             .RET_COUNT = ..RET_COUNT - .BLOCK_COUNT;
; 4896          6414  5             EXITLOOP;
; 4897          6415  5             END
; 4898          6416  5
; 4899          6417  4         ELSE
; 4900          6418  5             BEGIN
; 4901          6419  5             FREE_BLOCKS (.WP[WCB_COUNT], .WP[WCB_LBN]);
; 4902          6420  5             .RET_COUNT = ..RET_COUNT - .WP[WCB_COUNT];
; 4903          6421  5             BLOCK_COUNT = .BLOCK_COUNT - .WP[WCB_COUNT];
; 4904          6422  5             W[WCB_SIZE] = .W[WCB_SIZE] - 1;
; 4905          6423  5             IF .W[WCB_SIZE] EQL 0
; 4906          6424  5             AND .LAST_W NEQ 0
; 4907          6425  5             AND .W[WCB_RVN] EQL .LAST_W[WCB_RVN]
; 4908          6426  5             THEN
```

```
; 4909          6427  6                   BEGIN
; 4910          6428  6                   DELETE_WINDOW (.W);
; 4911          6429  6                   LAST_W[WCB_LINK] = 0;
; 4912          6430  5                   END;
; 4913          6431  4               END;
; 4914          6432  3           END;
; 4915          6433  3
; 4916          6434  3       ! Now build map pointers in the file header using the pointers in the
; 4917          6435  3       ! window.
; 4918          6436  3       !
; 4919          6437  3
; 4920          6438  3           W = .MTL[MTL_WINDOW];
; 4921          6439  3           UNTIL .W EQL 0
; 4922          6440  3           OR .W[WCB_RVN] EQL .CURRENT_VCB[VCB_RVN]
; 4923          6441  3           DO W = .W[WCB_LINK];
; 4924          6442  3
; 4925          6443  3           UNTIL .W EQL 0
; 4926          6444  3           DO
; 4927          6445  4               BEGIN
; 4928          6446  4               WP = .W + WCB_S_HEADER;
; 4929          6447  4               DECR J FROM .W[WCB_SIZE] TO 1
; 4930          6448  4               DO
; 4931          6449  5                   BEGIN
; 4932          6450  6                   STATUS = (IF .CURRENT_VCB[VCB_ODS_2]
; 4933          6451  6                             THEN MAKE_POINTER_E[SE_MAKE_POINTER1)
; 4934          6452  5                            (.MTL[MTL_READER], .WP[WCB_COUNT], .WP[WCB_LBN]);
; 4935          6453  5                   IF NOT .STATUS THEN RETURN .STATUS;
; 4936          6454  5                   WP = .WP + WCB_S_ENTRY;
; 4937          6455  4                   END;
; 4938          6456  4               W = .W[WCB_LINK];
; 4939          6457  3               END;
; 4940          6458  2           END;
; 4941          6459  2
; 4942          6460  2 TRUE
; 4943          6461  1 END;                                    ! End of routine STA_EXTEND


                                    01FC 00000          .ENTRY  STA_EXTEND, Save R2,R3,R4,R5,R6,R7,R8       ; 6314
                58 00000000'    EF  9E 00002        MOVAB   CURRENT_VCB, R8                             ;
                5E              08  C2 00009        SUBL2   #8, SP                                      ;
                50              68  D0 0000C        MOVL    CURRENT_VCB, R0                             ; 6366
                51          04  A0  3C 0000F        MOVZWL  4(R0), R1                                   ;
                51          04  AC  C0 00013        ADDL2   P_COUNT, R1                                 ;
                                51  D7 00017        DECL    R1                                          ;
                52          04  A0  3C 00019        MOVZWL  4(R0), R2                                   ; 6367
                51              52  C6 0001D        DIVL2   R2, R1                                      ;
                54          04  A0  3C 00020        MOVZWL  4(R0), BLOCK_COUNT                          ;
                54              51  C4 00024        MULL2   R1, BLOCK_COUNT                             ;
                            08  BC  D4 00027        CLRL    @RET_COUNT                                  ; 6369
                56          FC  A8  D0 0002A        MOVL    CURRENT_MTL, MTL                            ; 6370
                57          08  A6  D0 0002E        MOVL    8(MTL), R7                                  ; 6371
                                05  12 00032        BNEQ    1$                                          ;
                50          AC  8F  9A 00034        MOVZBL  #172, R0                                    ;
                                    04 00038        RET                                                 ;
```

```
                                    54  D5 00039 1$:    TSTL    BLOCK_COUNT                             ; 6377
                                    2D  15 0003B        BLEQ    3$                                      ;
                                    5E  DD 0003D        PUSHL   SP                                      ; 6380
                                08  AE  9F 0003F        PUSHAB  NEW_COUNT                               ;
                                    54  DD 00042        PUSHL   BLOCK_COUNT                             ;
                          DB69  CF      03  FB 00044    CALLS   #3, STA_ALLOC_BEST                      ;
                                06      50  E8 00049    BLBS    R0, 2$                                  ;
                                50  0850  8F  3C 0004C  MOVZWL  #2128, R0                              ; 6381
                                        04  00051        RET                                             ;
                                    6E  DD 00052 2$:    PUSHL   NEW_LBN                                 ; 6382
                                08  AE  DD 00054        PUSHL   NEW_COUNT                               ;
                                7E  1C  A6  9A 00057    MOVZBL  28(MTL), -(SP)                          ;
                                    57  DD 0005B        PUSHL   R7                                      ;
                          E05C  CF      04  FB 0005D    CALLS   #4, ADD_WINDOW_MAP                      ;
                            08  BC  04  AE  D0 00062    MOVL    NEW_COUNT, @RET_COUNT                   ; 6383
                                    00B9  31 00067      BRW     15$                                     ; 6377
                                54      54  CE 0006A 3$:    MNEGL   BLOCK_COUNT, BLOCK_COUNT            ; 6393
                                    54  D5 0006D 4$:    TSTL    BLOCK_COUNT                             ; 6394
                                    63  15 0006F        BLEQ    8$                                      ;
                                    55  D4 00071        CLRL    LAST_W                                  ; 6397
                                52      57  D0 00073    MOVL    R7, W                                   ; 6398
                                    62  D5 00076 5$:    TSTL    (W)                                     ; 6399
                                    08  13 00078        BEQL    6$                                      ;
                                55      52  D0 0007A    MOVL    W, LAST_W                               ; 6402
                                52      62  D0 0007D    MOVL    (W), W                                  ; 6403
                                    F4  11 00080        BRB     5$                                      ; 6399
                                08  A2  95 00082 6$:    TSTB    8(W)                                    ; 6405
                                    4D  13 00085        BEQL    8$                                      ;
                                50  08  A2  9A 00087    MOVZBL  8(W), R0                                ; 6407
                                53  0C A240  7E 0008B    MOVAQ   12(W)[R0], WP                           ;
                                54      63  D1 00090    CMPL    (WP), BLOCK_COUNT                       ; 6408
                                    15  15 00093        BLEQ    7$                                      ;
                                63      54  C2 00095    SUBL2   BLOCK_COUNT, (WP)                       ; 6411
                          7E    04  A3      63  C1 00098  ADDL3   (WP), 4(WP), -(SP)                    ; 6412
                                    54  DD 0009D        PUSHL   BLOCK_COUNT                             ;
                          DB9C  CF      02  FB 0009F    CALLS   #2, FREE_BLOCKS                         ;
                            08  BC      54  C2 000A4    SUBL2   BLOCK_COUNT, @RET_COUNT                 ; 6413
                                    2A  11 000A8        BRB     8$                                      ; 6410
                                7E      63  7D 000AA 7$:    MOVQ    (WP), -(SP)                         ; 6419
                          DB8E  CF      02  FB 000AD    CALLS   #2, FREE_BLOCKS                         ;
                            08  BC      63  C2 000B2    SUBL2   (WP), @RET_COUNT                        ; 6420
                                54      63  C2 000B6    SUBL2   (WP), BLOCK_COUNT                       ; 6421
                                    08  A2  97 000B9    DECB    8(W)                                    ; 6422
                                    AF  12 000BC        BNEQ    4$                                      ; 6423
                                    55  D5 000BE        TSTL    LAST_W                                  ; 6424
                                    AB  13 000C0        BEQL    4$                                      ;
                                0A  A5  0A  A2  91 000C2    CMPB    10(W), 10(LAST_W)                   ; 6425
                                    A4  12 000C7        BNEQ    4$                                      ;
                                    52  DD 000C9        PUSHL   W                                       ; 6428
                          DF73  CF      01  FB 000CB    CALLS   #1, DELETE_WINDOW                       ;
                                    65  D4 000D0        CLRL    (LAST_W)                                ; 6429
                                    99  11 000D2        BRB     4$                                      ; 6394
                                52      57  D0 000D4 8$:    MOVL    R7, W                               ; 6438
                                    0F  13 000D7 9$:    BEQL    10$                                     ; 6439
                                50      68  D0 000D9      MOVL    CURRENT_VCB, R0                         ; 6440
                                06  A0  0A  A2  91 000DC    CMPB    10(W), 6(R0)                        ;
                                    05  13 000E1        BEQL    10$                                     ;
```

```
                                52              62  D0 000E3          MOVL    (W), W                          ; 6441
                                                EF  11 000E6          BRB     9$                              ;
                                                52  D5 000E8 10$:     TSTL    W                               ; 6443
                                                37  13 000EA          BEQL    15$                             ;
                                53        14    A2  9E 000EC          MOVAB   20(R2), WP                      ; 6446
                                54        08    A2  9A 000F0          MOVZBL  8(W), J                         ; 6447
                                                54  D6 000F4          INCL    J                               ;
                                                23  11 000F6          BRB     14$                             ;
                                51              68  D0 000F8 11$:     MOVL    CURRENT_VCB, R1                 ; 6450
                      07        07    A1              01  E1 000FB          BBC     #1, 7(R1), 12$          ;
                                51        DC43  CF  9E 00100          MOVAB   MAKE_POINTER, R1                ;
                                                05  11 00105          BRB     13$                             ;
                                51        DBCB  CF  9E 00107 12$:     MOVAB   MAKE_POINTER1, R1               ;
                                7E              63  7D 0010C 13$:     MOVQ    (WP), -(SP)                     ; 6452
                                          0C    A6  DD 0010F          PUSHL   12(MTL)                         ;
                                61              03  FB 00112          CALLS   #3, (R1)                        ;
                                0E              50  E9 00115          BLBC    STATUS, 16$                     ; 6453
                                53              08  C0 00118          ADDL2   #8, WP                          ; 6454
                                DA              54  F5 0011B 14$:     SOBGTR  J, 11$                          ; 6447
                                52              62  D0 0011E          MOVL    (W), W                          ; 6456
                                                C5  11 00121          BRB     10$                             ; 6443
                                50              01  D0 00123 15$:     MOVL    #1, R0                          ; 6461
                                                    04 00126 16$:     RET                                     ;

; Routine Size:  295 bytes,    Routine Base:  CODE + 29A9
```

```
;  4945          6462  1 %SBTTL 'STA_RDWRVBLK - read/write virtual QIO routine'
;  4946          6463  1 ROUTINE STA_RDWRVBLK (EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)=
;  4947          6464  1
;  4948          6465  1 !++
;  4949          6466  1 !
;  4950          6467  1 ! FUNCTIONAL DESCRIPTION:
;  4951          6468  1 !       This routine executes IO$_READVBLK and IO$_WRITEVBLK in the
;  4952          6469  1 !       standalone environment.
;  4953          6470  1 !
;  4954          6471  1 ! INPUT PARAMETERS:
;  4955          6472  1 !       As for $QIO(W) system service.
;  4956          6473  1 !
;  4957          6474  1 ! IMPLICIT INPUTS:
;  4958          6475  1 !       CURRENT_MTL     - Pointer to MTL for selected volume set.
;  4959          6476  1 !
;  4960          6477  1 ! OUTPUT PARAMETERS:
;  4961          6478  1 !       NONE
;  4962          6479  1 !
;  4963          6480  1 ! IMPLICIT OUTPUTS:
;  4964          6481  1 !       NONE
;  4965          6482  1 !
;  4966          6483  1 ! ROUTINE VALUE:
;  4967          6484  1 !       Completion status.
;  4968          6485  1 !
;  4969          6486  1 ! SIDE EFFECTS:
;  4970          6487  1 !       NONE
;  4971          6488  1 !
;  4972          6489  1 !--
;  4973          6490  1
;  4974          6491  2 BEGIN
;  4975          6492  2 BUILTIN
;  4976          6493  2         AP;
;  4977          6494  2
;  4978          6495  2
;  4979          6496  2 ! Find the accessed file.  If none, error.
;  4980          6497  2 !
;  4981          6498  2 CURRENT_WCB = .CURRENT_MTL[MTL_WINDOW];
;  4982          6499  2 IF .CURRENT_WCB EQL 0 THEN RETURN SS$_FILNOTACC;
;  4983          6500  2
;  4984          6501  2
;  4985          6502  2 ! Do the I/O.
;  4986          6503  2 !
;  4987          6504  2 CALLG(.AP, R_W_VIRTUAL)
;  4988          6505  1 END;


                                    0000 00000 STA_RDWRVBLK:
                                                       .WORD   Save nothing                            ; 6463
                        50 00000000'    EF  D0 00002           MOVL    CURRENT_MTL, R0                 ; 6498
              00000000' EF        08    A0  D0 00009           MOVL    8(R0), CURRENT_WCB              ;
                                        05  12 00011           BNEQ    1$                              ; 6499
                        50        AC    8F  9A 00013           MOVZBL  #172, R0                        ;
                                        04  00017              RET                                     ;
                  DFFF  CF              6C  FA 00018 1$:       CALLG   (AP), R_W_VIRTUAL               ; 6504
```

```
                                04 0001D        RET                                     ; 6505

; Routine Size:  30 bytes,    Routine Base:  CODE + 2AD0
```

```
; 4990          6506  1 %SBTTL 'STA_ACCESS - access QIO routine'
; 4991          6507  1 ROUTINE STA_ACCESS (EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)=
; 4992          6508  1
; 4993          6509  1 !++
; 4994          6510  1 !
; 4995          6511  1 ! FUNCTIONAL DESCRIPTION:
; 4996          6512  1 !       This routine executes IO$_ACCESS in the standalone environment.
; 4997          6513  1 !
; 4998          6514  1 ! INPUT PARAMETERS:
; 4999          6515  1 !       As for $QIO(W) system service.
; 5000          6516  1 !
; 5001          6517  1 ! IMPLICIT INPUTS:
; 5002          6518  1 !       CURRENT_MTL     - Pointer to MTL for selected volume set.
; 5003          6519  1 !
; 5004          6520  1 ! OUTPUT PARAMETERS:
; 5005          6521  1 !       NONE
; 5006          6522  1 !
; 5007          6523  1 ! IMPLICIT OUTPUTS:
; 5008          6524  1 !       NONE
; 5009          6525  1 !
; 5010          6526  1 ! ROUTINE VALUE:
; 5011          6527  1 !       Completion status.
; 5012          6528  1 !
; 5013          6529  1 ! SIDE EFFECTS:
; 5014          6530  1 !       NONE
; 5015          6531  1 !
; 5016          6532  1 !--
; 5017          6533  1
; 5018          6534  2 BEGIN
; 5019          6535  2 MAP
; 5020          6536  2         FUNC:           BBLOCK,         ! I/O function code
; 5021          6537  2         P1:             REF BBLOCK;     ! Descriptor for FIB
; 5022          6538  2 LOCAL
; 5023          6539  2         FIB:            REF $BBLOCK,    ! Address of the FIB
; 5024          6540  2         LOCAL_HEADER:   BBLOCK[512],    ! Local header buffer
; 5025          6541  2         HEADER:         REF BBLOCK,     ! Pointer to header
; 5026          6542  2         STATUS,
; 5027          6543  2         VBN;                            ! Starting VBN of file
; 5028          6544  2
; 5029          6545  2
; 5030          6546  2
; 5031          6547  2 ! If necessary, access the specified file.  In the standalone ACP, the only
; 5032          6548  2 ! FIB field used is FIB$W_FID.
; 5033          6549  2 !
; 5034          6550  2 IF .P1 NEQ 0
; 5035          6551  2 THEN
; 5036          6552  3     BEGIN
; 5037          6553  3     FIB = .P1[DSC$A_POINTER];
; 5038          6554  3     IF .FUNC[IO$V_ACCESS]
; 5039          6555  3     THEN
; 5040          6556  4         BEGIN
; 5041          6557  4         !
; 5042          6558  4         ! Access specified:  check for file already accessed, save the file
; 5043          6559  4         ! ID, set up to use the MTL header buffer, read the header, and create
; 5044          6560  4         ! the window.
; 5045          6561  4         !
; 5046          6562  4         IF .CURRENT_MTL[MTL_WINDOW] NEQ 0 THEN RETURN SS$_FILALRACC;
```

```
;  5047          6563  4
;  5048          6564  4          QUEUE_HEADERS[0] = QUEUE_HEADERS[1] = QUEUE_HEADERS[0];
;  5049          6565  4          QUEUE_HEADERS[2] = QUEUE_HEADERS[3] = QUEUE_HEADERS[2];
;  5050          6566  4          QUEUE_HEADERS[4] = QUEUE_HEADERS[5] = QUEUE_HEADERS[4];
;  5051          6567  4          CURRENT_MTL[MTL_FID_NUM] = .FIB[FIB$W_FID_NUM];
;  5052          6568  4          CURRENT_MTL[MTL_FID_SEQ] = .FIB[FIB$W_FID_SEQ];
;  5053          6569  4          CURRENT_MTL[MTL_FID_RVNW] = .FIB[FIB$B_FID_RVN];
;  5054          6570  4          CURRENT_MTL[MTL_NEW_ACL] = 0;
;  5055          6571  4          HEADER = .CURRENT_MTL[MTL_HEADER];
;  5056          6572  4          STATUS = READ_HEADER(FIB[FIB$W_FID_NUM], .HEADER);
;  5057          6573  4          IF NOT .STATUS THEN RETURN .STATUS;
;  5058          6574  4          VBN = 1;
;  5059          6575  4          IF .P1[DSC$W_LENGTH] GEQU FIB$C_EXTDATA
;  5060          6576  4          AND .FIB[FIB$L_EXVBN] NEQ 0
;  5061          6577  4          THEN VBN = .FIB[FIB$L_EXVBN];
;  5062          6578  4          STATUS = CREATE_WINDOW(.HEADER, .FIB[FIB$B_FID_RVN],
;  5063          6579  4                   CURRENT_MTL[MTL_WINDOW], .VBN, .FIB[FIB$B_WSIZE]);
;  5064          6580  4          IF NOT .STATUS THEN RETURN .STATUS;
;  5065          6581  4          END
;  5066          6582  3      ELSE
;  5067          6583  4          BEGIN
;  5068          6584  4          !
;  5069          6585  4          ! No access specified:  allow file already accessed, use the local
;  5070          6586  4          ! header buffer to avoid destroying context, and read the header.
;  5071          6587  4          !
;  5072          6588  4          HEADER = LOCAL_HEADER;
;  5073          6589  4          STATUS = READ_HEADER(FIB[FIB$W_FID_NUM], .HEADER);
;  5074          6590  4          IF NOT .STATUS THEN RETURN .STATUS;
;  5075          6591  3          END;
;  5076          6592  3      END
;  5077          6593  2  ELSE
;  5078          6594  3      BEGIN
;  5079          6595  3      IF .CURRENT_MTL[MTL_WINDOW] EQL 0 THEN RETURN SS$_BADPARAM;
;  5080          6596  3      HEADER = .CURRENT_MTL[MTL_HEADER];
;  5081          6597  2      END;
;  5082          6598  2
;  5083          6599  2
;  5084          6600  2  ! If necessary, build the file's ACL.
;  5085          6601  2  !
;  5086          6602  2  IF .CURRENT_MTL[MTL_ACLFL] EQL 0
;  5087          6603  2  THEN
;  5088          6604  3      BEGIN
;  5089          6605  3      CURRENT_MTL[MTL_ACLFL] = CURRENT_MTL[MTL_ACLFL];     ! Initialize ACL
;  5090          6606  3      CURRENT_MTL[MTL_ACLBL] = CURRENT_MTL[MTL_ACLFL];     !   queue head
;  5091          6607  3      IF NOT .CURRENT_MTL[MTL_SEQ_DISK]
;  5092          6608  3      THEN
;  5093          6609  4          BEGIN
;  5094          6610  4          STATUS = ACL_BUILDACL ();
;  5095          6611  4          IF NOT .STATUS
;  5096          6612  4          THEN
;  5097          6613  5              BEGIN
;  5098          6614  5              IF .FUNC[IO$V_ACCESS]
;  5099          6615  5              THEN
;  5100          6616  6                  BEGIN
;  5101          6617  6                  ACL_DELETEACL ();
;  5102          6618  6                  CURRENT_MTL[MTL_ACLFL] = CURRENT_MTL[MTL_ACLBL] = 0;
;  5103          6619  6                  DELETE_WINDOW (.CURRENT_MTL[MTL_WINDOW]);
```

```
;  5104          6620  6                    CURRENT_MTL[MTL_WINDOW] = 0;
;  5105          6621  5                    END;
;  5106          6622  5                RETURN .STATUS;
;  5107          6623  4                END;
;  5108          6624  3            END;
;  5109          6625  2        END;
;  5110          6626  2
;  5111          6627  2
;  5112          6628  2    ! If specified, read attributes.
;  5113          6629  2    !
;  5114          6630  2    IF .P5 NEQ 0
;  5115          6631  2    THEN
;  5116          6632  3        BEGIN
;  5117          6633  3        STATUS = READ_ATTRIBUTES(.HEADER, .P5, (IF .P1 EQL 0 THEN 0 ELSE .P1[DSC$A_POINTER]));
;  5118          6634  3        IF NOT .STATUS
;  5119          6635  3        THEN
;  5120          6636  4            BEGIN
;  5121          6637  4            IF .FUNC[IO$V_ACCESS]
;  5122          6638  4            THEN
;  5123          6639  5                BEGIN
;  5124          6640  5                ACL_DELETEACL ();
;  5125          6641  5                CURRENT_MTL[MTL_ACLFL] = CURRENT_MTL[MTL_ACLBL] = 0;
;  5126          6642  5                DELETE_WINDOW (.CURRENT_MTL[MTL_WINDOW]);
;  5127          6643  5                CURRENT_MTL[MTL_WINDOW] = 0;
;  5128          6644  4                END;
;  5129          6645  4            RETURN .STATUS;
;  5130          6646  3            END;
;  5131          6647  2        END;
;  5132          6648  2
;  5133          6649  2
;  5134          6650  2    ! Completed normally.
;  5135          6651  2    !
;  5136          6652  2    SS$_NORMAL
;  5137          6653  1    END;


                                    00FC 00000 STA_ACCESS:
                                                       .WORD   Save R2,R3,R4,R5,R6,R7                  ; 6507
                                57 00000000'   EF  9E 00002    MOVAB   CURRENT_MTL, R7                 ;
                                56 00000000'   EF  9E 00009    MOVAB   QUEUE_HEADERS, R6               ;
                                5E     FE00    CE  9E 00010    MOVAB   -512(SP), SP                    ;
                                53       1C    AC  D0 00015    MOVL    P1, R3                          ; 6550
                                               03  12 00019    BNEQ    1$                              ;
                                             0097  31 0001B    BRW     6$                              ;
                                52       04    A3  D0 0001E 1$: MOVL    4(R3), FIB                      ; 6553
                          79    0C    AC        06  E1 00022    BBC     #6, FUNC, 4$                    ; 6554
                                50             67  D0 00027    MOVL    CURRENT_MTL, R0                 ; 6562
                                         08    A0  D5 0002A    TSTL    8(R0)                           ;
                                               05  13 0002D    BEQL    2$                              ;
                                50       A4    8F  9A 0002F    MOVZBL  #164, R0                        ;
                                               04  00033    RET                                     ;
                                51             66  9E 00034 2$: MOVAB   QUEUE_HEADERS, R1              ; 6564
                                   04    A6        51  D0 00037    MOVL    R1, QUEUE_HEADERS+4             ;
                                66             51  D0 0003B    MOVL    R1, QUEUE_HEADERS               ;
```

```
                                51      08      A6  9E 0003E       MOVAB   QUEUE_HEADERS+8, R1              ; 6565
                            0C  A6              51  D0 00042       MOVL    R1, QUEUE_HEADERS+12            ;
                            08  A6              51  D0 00046       MOVL    R1, QUEUE_HEADERS+8             ;
                                51      10      A6  9E 0004A       MOVAB   QUEUE_HEADERS+16, R1            ; 6566
                            14  A6              51  D0 0004E       MOVL    R1, QUEUE_HEADERS+20            ;
                            10  A6              51  D0 00052       MOVL    R1, QUEUE_HEADERS+16            ;
                            18  A0      04      A2  D0 00056       MOVL    4(FIB), 24(R0)                  ; 6567
                            1C  A0      08      A2  B0 0005B       MOVW    8(FIB), 28(R0)                  ; 6569
                            31  A0              02  8A 00060       BICB2   #2, 49(R0)                      ; 6570
                                55      0C      A0  D0 00064       MOVL    12(R0), HEADER                  ; 6571
                                                55  DD 00068       PUSHL   HEADER                          ; 6572
                                        04      A2  9F 0006A       PUSHAB  4(FIB)                          ;
                          D750  CF              02  FB 0006D       CALLS   #2, READ_HEADER                 ;
                                54              50  D0 00072       MOVL    R0, STATUS                      ;
                                75              54  E9 00075       BLBC    STATUS, 9$                      ; 6573
                                50              01  D0 00078       MOVL    #1, VBN                         ; 6574
                                20              63  B1 0007B       CMPW    (R3), #32                       ; 6575
                                                09  1F 0007E       BLSSU   3$                              ;
                                        1C      A2  D5 00080       TSTL    28(FIB)                         ; 6576
                                                04  13 00083       BEQL    3$                              ;
                                50      1C      A2  D0 00085       MOVL    28(FIB), VBN                    ; 6577
                                7E      03      A2  98 00089 3$:   CVTBL   3(FIB), -(SP)                   ; 6579
                                                50  DD 0008D       PUSHL   VBN                             ;
                          7E                    67  08  C1 0008F   ADDL3   #8, CURRENT_MTL, -(SP)          ;
                                7E      08      A2  9A 00093       MOVZBL  8(FIB), -(SP)                   ;
                                                55  DD 00097       PUSHL   HEADER                          ;
                          DC42  CF              05  FB 00099       CALLS   #5, CREATE_WINDOW               ;
                                                0D  11 0009E       BRB     5$                              ;
                                55              6E  9E 000A0 4$:   MOVAB   LOCAL_HEADER, HEADER            ; 6588
                                                55  DD 000A3       PUSHL   HEADER                          ; 6589
                                        04      A2  9F 000A5       PUSHAB  4(FIB)                          ;
                          D715  CF              02  FB 000A8       CALLS   #2, READ_HEADER                 ;
                                54              50  D0 000AD 5$:   MOVL    R0, STATUS                      ;
                                12              54  E8 000B0       BLBS    STATUS, 8$                      ; 6590
                                                7A  11 000B3       BRB     14$                             ;
                                50              67  D0 000B5 6$:   MOVL    CURRENT_MTL, R0                 ; 6595
                                        08      A0  D5 000B8       TSTL    8(R0)                           ;
                                                04  12 000BB       BNEQ    7$                              ;
                                50              14  D0 000BD       MOVL    #20, R0                         ;
                                                04  000C0         RET                                     ;
                                55      0C      A0  D0 000C1 7$:   MOVL    12(R0), HEADER                  ; 6596
                                50              67  D0 000C5 8$:   MOVL    CURRENT_MTL, R0                 ; 6602
                                        10      A0  D5 000C8       TSTL    16(R0)                          ;
                                                22  12 000CB       BNEQ    10$                             ;
                            10  A0      10      A0  9E 000CD       MOVAB   16(R0), 16(R0)                  ; 6605
                            14  A0      10      A0  9E 000D2       MOVAB   16(R0), 20(R0)                  ; 6606
                                14      31      A0  E8 000D7       BLBS    49(R0), 10$                     ; 6607
                    00000000G   00              00  FB 000DB       CALLS   #0, ACL_BUILDACL                ; 6610
                                54              50  D0 000E2       MOVL    R0, STATUS                      ;
                                07              54  E8 000E5       BLBS    STATUS, 10$                     ; 6611
                          27    0C      AC              06  E0 000E8   BBS     #6, FUNC, 13$               ; 6614
                                                40  11 000ED 9$:   BRB     14$                             ; 6622
                                        2C      AC  D5 000EF 10$:  TSTL    P5                              ; 6630
                                                3F  13 000F2       BEQL    15$                             ;
                                                53  D5 000F4       TSTL    R3                              ; 6633
                                                04  12 000F6       BNEQ    11$                             ;
                                                7E  D4 000F8       CLRL    -(SP)                           ;
```

```
                                        03  11 000FA          BRB     12$                                                     ;
                                    04  A3  DD 000FC 11$:     PUSHL   4(R3)                                                   ;
                                    2C  AC  DD 000FF 12$:     PUSHL   P5                                                      ;
                                        55  DD 00102          PUSHL   HEADER                                                  ;
                              0000V CF      03  FB 00104      CALLS   #3, READ_ATTRIBUTES                                     ;
                                    54      50  D0 00109      MOVL    R0, STATUS                                              ;
                                    24      54  E8 0010C      BLBS    STATUS, 15$                                             ; 6634
                          1B        0C  AC  06  E1 0010F      BBC     #6, FUNC, 14$                                           ; 6637
                             00000000G  00      00  FB 00114 13$:     CALLS   #0, ACL_DELETEACL                               ; 6640
                                    50      67  D0 0011B      MOVL    CURRENT_MTL, R0                                         ; 6641
                                    10  A0  7C 0011E          CLRQ    16(R0)                                                  ;
                                    08  A0  DD 00121          PUSHL   8(R0)                                                   ; 6642
                              DDD5  CF      01  FB 00124      CALLS   #1, DELETE_WINDOW                                       ;
                                    50      67  D0 00129      MOVL    CURRENT_MTL, R0                                         ; 6643
                                    08  A0  D4 0012C          CLRL    8(R0)                                                   ;
                                    50      54  D0 0012F 14$.     MOVL    STATUS, R0                                          ; 6645
                                            04  00132          RET                                                             ;
                                    50      01  D0 00133 15$:     MOVL    #1, R0                                              ; 6653
                                            04  00136          RET                                                             ;

; Routine Size:  311 bytes,    Routine Base:  CODE + 2AEE
```

```
;  5139     6654  1 %SBTTL 'CREATE_CLEANUP - clean up after create failure'
;  5140     6655  1 ROUTINE CREATE_CLEANUP (CREATE_STATUS)=
;  5141     6656  1
;  5142     6657  1 !++
;  5143     6658  1 !
;  5144     6659  1 ! FUNCTIONAL DESCRIPTION:
;  5145     6660  1 !       This routine processes the extent list produced during an IO$_CREATE
;  5146     6661  1 !       and frees the disk extents if the create failed.  In any case, the
;  5147     6662  1 !       extent list, the create list, and the window are freed.
;  5148     6663  1 !
;  5149     6664  1 ! INPUT PARAMETERS:
;  5150     6665  1 !       CREATE_STATUS   - Create status.
;  5151     6666  1 !
;  5152     6667  1 ! IMPLICIT INPUTS:
;  5153     6668  1 !       NONE
;  5154     6669  1 !
;  5155     6670  1 ! OUTPUT PARAMETERS:
;  5156     6671  1 !       NONE
;  5157     6672  1 !
;  5158     6673  1 ! IMPLICIT OUTPUTS:
;  5159     6674  1 !       NONE
;  5160     6675  1 !
;  5161     6676  1 ! ROUTINE VALUE:
;  5162     6677  1 !       The create status.
;  5163     6678  1 !
;  5164     6679  1 ! SIDE EFFECTS:
;  5165     6680  1 !       NONE
;  5166     6681  1 !
;  5167     6682  1 !--
;  5168     6683  1
;  5169     6684  2 BEGIN
;  5170     6685  2 LOCAL
;  5171     6686  2         P:              REF BBLOCK;     ! Pointer to list entry
;  5172     6687  2
;  5173     6688  2
;  5174     6689  2 ! Free the extent list.
;  5175     6690  2 !
;  5176     6691  2 UNTIL REMQUE(.QUEUE_HEADERS[0], P) DO
;  5177     6692  3     BEGIN
;  5178     6693  3
;  5179     6694  3     ! If the create failed, release the allocated disk blocks.
;  5180     6695  3     !
;  5181     6696  3     IF NOT .CREATE_STATUS AND .P[EXT_COUNT] NEQ 0
;  5182     6697  3     THEN
;  5183     6698  4         BEGIN
;  5184     6699  4         CURRENT_VCB = .P[EXT_VCB];
;  5185     6700  4         FREE_BLOCKS(.P[EXT_COUNT], .P[EXT_LBN]);
;  5186     6701  3         END;
;  5187     6702  3
;  5188     6703  3
;  5189     6704  3     ! Deallocate the extent list entry.
;  5190     6705  3     !
;  5191     6706  3     FREE_VM(EXT_S_ENTRY, .P);
;  5192     6707  2     END;
;  5193     6708  2
;  5194     6709  2
;  5195     6710  2 ! Free the create list if any errors occurred.
```

```
; 5196          6711  2 !
; 5197          6712  2 IF NOT .CREATE_STATUS
; 5198          6713  2 THEN
; 5199          6714  2     UNTIL REMQUE(.QUEUE_HEADERS[2], P) DO
; 5200          6715  3         BEGIN
; 5201          6716  3         LOCAL
; 5202          6717  3             Q:          REF BBLOCK;     ! Pointer to FID block
; 5203          6718  3
; 5204          6719  3
; 5205          6720  3         ! Free the FID blocks.
; 5206          6721  3         !
; 5207          6722  3         UNTIL REMQUE(.P[CRT_FID_FQHDR], Q) DO FREE_VM(CRT_S_FID, .Q);
; 5208          6723  3
; 5209          6724  3
; 5210          6725  3         ! Deallocate the create list entry.
; 5211          6726  3         !
; 5212          6727  3         FREE_VM(CRT_S_BLOCKS, .P);
; 5213          6728  2         END;
; 5214          6729  2
; 5215          6730  2
; 5216          6731  2 ! Free the used file ID list.
; 5217          6732  2 !
; 5218          6733  2 UNTIL REMQUE(.QUEUE_HEADERS[4], P) DO
; 5219          6734  3     BEGIN
; 5220          6735  3     LOCAL
; 5221          6736  3         HEADER:         BBLOCK[512];    ! File header buffer
; 5222          6737  3
; 5223          6738  3
; 5224          6739  3     ! If the create failed, write a deleted file header.
; 5225          6740  3     !
; 5226          6741  3     IF NOT .CREATE_STATUS
; 5227          6742  3     THEN
; 5228          6743  4         BEGIN
; 5229          6744  4         CREATE_DELHDR(P[CRT_FID], HEADER);
; 5230          6745  4         WRITE_HEADER(P[CRT_FID], HEADER);
; 5231          6746  3         END;
; 5232          6747  3
; 5233          6748  3
; 5234          6749  3     ! Deallocate the create list entry.
; 5235          6750  3     !
; 5236          6751  3     FREE_VM(CRT_S_FID, .P);
; 5237          6752  2     END;
; 5238          6753  2
; 5239          6754  2
; 5240          6755  2 ! Release the window if one exists.
; 5241          6756  2 !
; 5242          6757  2 IF NOT .CREATE_STATUS AND .CURRENT_MTL[MTL_WINDOW] NEQ 0
; 5243          6758  2 THEN
; 5244          6759  3     BEGIN
; 5245          6760  3     DELETE_WINDOW(.CURRENT_MTL[MTL_WINDOW]);
; 5246          6761  3     CURRENT_MTL[MTL_WINDOW] = 0;
; 5247          6762  2     END;
; 5248          6763  2
; 5249          6764  2
; 5250          6765  2 ! Return the original status.
; 5251          6766  2 !
; 5252          6767  2 .CREATE_STATUS
```

; 5253 6768 1 END;

```
                                 007C 00000 CREATE_CLEANUP:
                                            .WORD   Save R2,R3,R4,R5,R6                 ; 6655
                    56 00000000'  EF  9E 00002          MOVAB   QUEUE_HEADERS, R6       ;
                    55 00000000G  00  9E 00009          MOVAB   FREE_VM, R5             ;
                    54 00000000'  EF  9E 00010          MOVAB   CURRENT_MTL, R4         ;
                    5E     FE00   CE  9E 00017          MOVAB   -512(SP), SP            ;
                    52       00   B6  0F 0001C 1$:      REMQUE  @QUEUE_HEADERS, P       ; 6691
                                  20  1D 00020          BVS     3$                      ;
                    13       04   AC  E8 00022          BLBS    CREATE_STATUS, 2$       ; 6696
                             0C   A2  D5 00026          TSTL    12(P)                   ;
                                  0E  13 00029          BEQL    2$                      ;
                 04 A4       08   A2  D0 0002B          MOVL    8(P), CURRENT_VCB       ; 6699
                 7E          0C   A2  7D 00030          MOVQ    12(P), -(SP)            ; 6700
              D98B CF                 02  FB 00034          CALLS   #2, FREE_BLOCKS         ;
                                      52  DD 00039 2$:      PUSHL   P                       ; 6706
                                      14  DD 0003B          PUSHL   #20                     ;
                    65                02  FB 0003D          CALLS   #2, FREE_VM             ;
                                      DA  11 00040          BRB     1$                      ; 6691
                    1E       04   AC  E8 00042 3$:      BLBS    CREATE_STATUS, 7$       ; 6712
                    52       08   B6  0F 00046 4$:      REMQUE  @QUEUE_HEADERS+8, P     ; 6714
                                  18  1D 0004A          BVS     7$                      ;
                    53       08   B2  0F 0004C 5$:      REMQUE  @8(P), Q                ; 6722
                                  09  1D 00050          BVS     6$                      ;
                                      53  DD 00052          PUSHL   Q                       ;
                                      0E  DD 00054          PUSHL   #14                     ;
                    65                02  FB 00056          CALLS   #2, FREE_VM             ;
                                      F1  11 00059          BRB     5$                      ;
                                      52  DD 0005B 6$:      PUSHL   P                       ; 6727
                                      14  DD 0005D          PUSHL   #20                     ;
                    65                02  FB 0005F          CALLS   #2, FREE_VM             ;
                                      E2  11 00062          BRB     4$                      ; 6714
                    52       10   B6  0F 00064 7$:      REMQUE  @QUEUE_HEADERS+16, P    ; 6733
                                  21  1D 00068          BVS     9$                      ;
                    14       04   AC  E8 0006A          BLBS    CREATE_STATUS, 8$       ; 6741
                                      5E  DD 0006E          PUSHL   SP                      ; 6744
                             08   A2  9F 00070          PUSHAB  8(P)                    ;
              D795 CF                 02  FB 00073          CALLS   #2, CREATE_DELHDR       ;
                                      5E  DD 00078          PUSHL   SP                      ; 6745
                             08   A2  9F 0007A          PUSHAB  8(P)                    ;
              D699 CF                 02  FB 0007D          CALLS   #2, WRITE_HEADER        ;
                                      52  DD 00082 8$:      PUSHL   P                       ; 6751
                                      0E  DD 00084          PUSHL   #14                     ;
                    65                02  FB 00086          CALLS   #2, FREE_VM             ;
                                      D9  11 00089          BRB     7$                      ; 6733
                    19       04   AC  E8 0008B 9$:      BLBS    CREATE_STATUS, 10$      ; 6757
                    50                64  D0 0008F          MOVL    CURRENT_MTL, R0         ;
                             08   A0  D5 00092          TSTL    8(R0)                   ;
                                  11  13 00095          BEQL    10$                     ;
                    50                64  D0 00097          MOVL    CURRENT_MTL, R0         ; 6760
                             08   A0  DD 0009A          PUSHL   8(R0)                   ;
              DD25 CF                 01  FB 0009D          CALLS   #1, DELETE_WINDOW       ;
```

```
                                50              64  D0 000A2            MOVL    CURRENT_MTL, R0                 ; 6761
                                                    D4 000A5            CLRL    8(R0)                           ;
                                50          08  AC  D0 000A8 10$:      MOVL    CREATE_STATUS, R0               ; 6768
                                                    04 000AC            RET                                     ;

; Routine Size:  173 bytes,    Routine Base:  CODE + 2C25
```

```
;  5255          6769  1 %SBTTL 'CREATE_EXTHDR - create extension header'
;  5256          6770  1 ROUTINE CREATE_EXTHDR (OLD_HDR,OLD_FILE_ID,NEW_HDR,NEW_FILE_ID)=
;  5257          6771  1
;  5258          6772  1 !++
;  5259          6773  1 !
;  5260          6774  1 ! FUNCTIONAL DESCRIPTION:
;  5261          6775  1 !       This routine creates an extension header.
;  5262          6776  1 !
;  5263          6777  1 ! INPUT PARAMETERS:
;  5264          6778  1 !       OLD_HDR         - Pointer to current file header.
;  5265          6779  1 !       OLD_FILE_ID     - File ID of current file header.
;  5266          6780  1 !       NEW_HDR         - Pointer to buffer where extension header is built.
;  5267          6781  1 !       NEW_FILE_ID     - File ID of extension file header.
;  5268          6782  1 !
;  5269          6783  1 ! IMPLICIT INPUTS:
;  5270          6784  1 !       CURRENT_MTL     - Pointer to MTL for current volume set.
;  5271          6785  1 !
;  5272          6786  1 ! OUTPUT PARAMETERS:
;  5273          6787  1 !       NONE
;  5274          6788  1 !
;  5275          6789  1 ! IMPLICIT OUTPUTS:
;  5276          6790  1 !       NONE
;  5277          6791  1 !
;  5278          6792  1 ! ROUTINE VALUE:
;  5279          6793  1 !       Completion status.
;  5280          6794  1 !
;  5281          6795  1 ! SIDE EFFECTS:
;  5282          6796  1 !       NONE
;  5283          6797  1 !
;  5284          6798  1 !--
;  5285          6799  1
;  5286          6800  2 BEGIN
;  5287          6801  2 MAP
;  5288          6802  2         OLD_HDR:        REF BBLOCK,     ! Pointer to file header
;  5289          6803  2         OLD_FILE_ID:    REF BBLOCK,     ! Pointer to file ID
;  5290          6804  2         NEW_HDR:        REF BBLOCK,     ! Pointer to file header
;  5291          6805  2         NEW_FILE_ID:    REF BBLOCK;     ! Pointer to file ID
;  5292          6806  2
;  5293          6807  2
;  5294          6808  2 IF .OLD_HDR[FH2$B_STRUCLEV] EQL 2
;  5295          6809  2 THEN
;  5296          6810  3     BEGIN
;  5297          6811  3
;  5298          6812  3     ! Make sure the segment number will not overflow.
;  5299          6813  3     !
;  5300          6814  3     IF .OLD_HDR[FH2$W_SEG_NUM] GEQU 65535
;  5301          6815  3     THEN
;  5302          6816  3         RETURN SS$_HEADERFULL;
;  5303          6817  3
;  5304          6818  3
;  5305          6819  3     ! Put the extension linkage in the previous header.
;  5306          6820  3     !
;  5307          6821  3     OLD_HDR[FH2$W_EX_FIDNUM] = .NEW_FILE_ID[FID$W_NUM];
;  5308          6822  3     OLD_HDR[FH2$W_EX_FIDSEQ] = .NEW_FILE_ID[FID$W_SEQ];
;  5309          6823  3     OLD_HDR[FH2$W_EX_FIDRVN] = .NEW_FILE_ID[FID$W_RVN];
;  5310          6824  3     IF .OLD_FILE_ID[FID$B_RVN] EQL .NEW_FILE_ID[FID$B_RVN]
;  5311          6825  3         THEN OLD_HDR[FH2$B_EX_FIDRVN] = 0;
```

```
;  5312          6826  3     END
;  5313          6827  2 ELSE
;  5314          6828  3     BEGIN
;  5315          6829  3     LOCAL
;  5316          6830  3         MAP_AREA:       REF BBLOCK;     ! Pointer to map area
;  5317          6831  3
;  5318          6832  3
;  5319          6833  3     MAP_AREA = .OLD_HDR + .OLD_HDR[FH1$B_MPOFFSET]*2;
;  5320          6834  3
;  5321          6835  3
;  5322          6836  3     ! Make sure the segment number will not overflow.
;  5323          6837  3     !
;  5324          6838  3     IF .MAP_AREA[FM1$B_EX_SEGNUM] GEQU 255
;  5325          6839  3     THEN
;  5326          6840  3         RETURN SS$_HEADERFULL;
;  5327          6841  3
;  5328          6842  3
;  5329          6843  3     ! Put the extension linkage in the previous header.
;  5330          6844  3     !
;  5331          6845  3     MAP_AREA[FM1$W_EX_FILNUM] = .NEW_FILE_ID[FID$W_NUM];
;  5332          6846  3     MAP_AREA[FM1$W_EX_FILSEQ] = .NEW_FILE_ID[FID$W_SEQ];
;  5333          6847  2     END;
;  5334          6848  2
;  5335          6849  2
;  5336          6850  2 ! If the old header and the new header occupy the same buffer, the old header
;  5337          6851  2 ! is an extension header.  Otherwise, it is the primary header.  Write the old
;  5338          6852  2 ! header if necessary.
;  5339          6853  2 !
;  5340          6854  2 IF .OLD_HDR EQL .NEW_HDR
;  5341          6855  2 THEN
;  5342          6856  3     BEGIN
;  5343          6857  3     LOCAL
;  5344          6858  3         STATUS;                         ! Status variable
;  5345          6859  3
;  5346          6860  3     IF
;  5347          6861  3         NOT .QUAL[QUAL_VOLU] OR
;  5348          6862  3         .QUAL[QUAL_VOLO_VALUE] EQL .OLD_FILE_ID[FID$B_RVN]
;  5349          6863  3     THEN
;  5350          6864  4         BEGIN
;  5351          6865  4         STATUS = WRITE_HEADER(.OLD_FILE_ID, .OLD_HDR);
;  5352          6866  4         IF NOT .STATUS THEN RETURN .STATUS;
;  5353          6867  3         END;
;  5354          6868  3     END
;  5355          6869  2 ELSE
;  5356          6870  2     CH$MOVE(512, .OLD_HDR, .NEW_HDR);
;  5357          6871  2
;  5358          6872  2
;  5359          6873  2 IF .NEW_HDR[FH2$B_STRUCLEV] EQL 2
;  5360          6874  2 THEN
;  5361          6875  3     BEGIN
;  5362          6876  3
;  5363          6877  3     ! Place the file ID in the new header.
;  5364          6878  3     !
;  5365          6879  3     NEW_HDR[FH2$W_FID_NUM] = .NEW_FILE_ID[FID$W_NUM];
;  5366          6880  3     NEW_HDR[FH2$W_FID_SEQ] = .NEW_FILE_ID[FID$W_SEQ];
;  5367          6881  3     NEW_HDR[FH2$B_FID_RVN] = 0;
;  5368          6882  3     NEW_HDR[FH2$B_FID_NMX] = .NEW_FILE_ID[FID$B_NMX];
```

```
;  5369          6883  3
;  5370          6884  3
;  5371          6885  3      ! Set the extension header back link to point to the primary header.
;  5372          6886  3      !
;  5373          6887  3      NEW_HDR[FH2$W_BK_FIDNUM] = .CURRENT_MTL[MTL_FID_NUM];
;  5374          6888  3      NEW_HDR[FH2$W_BK_FIDSEQ] = .CURRENT_MTL[MTL_FID_SEQ];
;  5375          6889  3      NEW_HDR[FH2$W_BK_FIDRVN] = .CURRENT_MTL[MTL_FID_RVNW];
;  5376          6890  3      IF .NEW_HDR[FH2$B_BK_FIDRVN] EQL .NEW_FILE_ID[FID$B_RVN]
;  5377          6891  3          THEN NEW_HDR[FH2$B_BK_FIDRVN] = 0;
;  5378          6892  3
;  5379          6893  3
;  5380          6894  3      ! Finish initializing the new header.
;  5381          6895  3      !
;  5382          6896  3      NEW_HDR[FH2$W_SEG_NUM] = .NEW_HDR[FH2$W_SEG_NUM] + 1;
;  5383          6897  3      NEW_HDR[FH2$W_EX_FIDNUM] = 0;
;  5384          6898  3      NEW_HDR[FH2$W_EX_FIDSEQ] = 0;
;  5385          6899  3      NEW_HDR[FH2$W_EX_FIDRVN] = 0;
;  5386          6900  3      NEW_HDR[FH2$B_MAP_INUSE] = 0;
;  5387          6901  3      NEW_HDR[FH2$V_CONTIG] = FALSE;
;  5388          6902  3      NEW_HDR[FH2$V_LOCKED] = FALSE;
;  5389          6903  3      BBLOCK[NEW_HDR[FH2$W_RECATTR], FAT$L_HIBLK] = 0;
;  5390          6904  3      BBLOCK[NEW_HDR[FH2$W_RECATTR], FAT$L_EFBLK] = 0;
;  5391          6905  3      BBLOCK[NEW_HDR[FH2$W_RECATTR], FAT$W_FFBYTE] = 0;
;  5392          6906  3
;  5393          6907  3
;  5394          6908  3      ! Truncate the ident area to provide more space for the map area.
;  5395          6909  3      ! Clear the map area.
;  5396          6910  3      !
;  5397          6911  3      NEW_HDR[FH2$B_MPOFFSET] = .NEW_HDR[FH2$B_IDOFFSET] + $BYTEOFFSET(FI2$W_REVISION) / 2;
;  5398          6912  3      NEW_HDR[FH2$B_ACOFFSET] = $BYTEOFFSET(FH2$W_CHECKSUM) / 2;
;  5399          6913  3      NEW_HDR[FH2$B_RSOFFSET] = $BYTEOFFSET(FH2$W_CHECKSUM) / 2;
;  5400          6914  3      CH$FILL(0, 512 - .NEW_HDR[FH2$B_MPOFFSET]*2, .NEW_HDR + .NEW_HDR[FH2$B_MPOFFSET]*2);
;  5401          6915  3      END
;  5402          6916  2 ELSE
;  5403          6917  3      BEGIN
;  5404          6918  3      LOCAL
;  5405          6919  3          MAP_AREA:       REF BBLOCK;     ! Pointer to map area
;  5406          6920  3
;  5407          6921  3
;  5408          6922  3      MAP_AREA = .NEW_HDR + .NEW_HDR[FH1$B_MPOFFSET] * 2;
;  5409          6923  3
;  5410          6924  3
;  5411          6925  3      ! Place the file ID in the new header.
;  5412          6926  3      !
;  5413          6927  3      NEW_HDR[FH1$W_FID_NUM] = .NEW_FILE_ID[FID$W_NUM];
;  5414          6928  3      NEW_HDR[FH1$W_FID_SEQ] = .NEW_FILE_ID[FID$W_SEQ];
;  5415          6929  3
;  5416          6930  3
;  5417          6931  3      ! Finish initializing the new header.
;  5418          6932  3      !
;  5419          6933  3      BBLOCK[NEW_HDR[FH1$W_RECATTR], FAT$L_HIBLK] = 0;
;  5420          6934  3      BBLOCK[NEW_HDR[FH1$W_RECATTR], FAT$L_EFBLK] = 0;
;  5421          6935  3      BBLOCK[NEW_HDR[FH1$W_RECATTR], FAT$W_FFBYTE] = 0;
;  5422          6936  3      MAP_AREA[FM1$B_EX_SEGNUM] = .MAP_AREA[FM1$B_EX_SEGNUM] + 1;
;  5423          6937  3      MAP_AREA[FM1$W_EX_FILNUM] = 0;
;  5424          6938  3      MAP_AREA[FM1$W_EX_FILSEQ] = 0;
;  5425          6939  3      MAP_AREA[FM1$B_INUSE] = 0;
```

```
;  5426          6940  3     CH$FILL(0, .MAP_AREA[FM1$B_AVAIL] * 2, .MAP_AREA + FM1$C_POINTERS);
;  5427          6941  2     END;
;  5428          6942  2
;  5429          6943  2
;  5430          6944  2 ! Indicate success.
;  5431          6945  2 !
;  5432          6946  2 SS$_NORMAL
;  5433          6947  1 END;


                                                07FC 00000 CREATE_EXTHDR:
                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10                ; 6770
                          5A 00000000'  EF  9E 00002       MOVAB   QUAL+12, R10
                          57        10  AC  D0 00009       MOVL    NEW_FILE_ID, R7                                ; 6821
                          50        04  AC  D0 0000D       MOVL    OLD_HDR, R0                                    ; 6808
                          02        07  A0  91 00011       CMPB    7(R0), #2
                                        2C  12 00015       BNEQ    1$
                      FFFF  8F      04  A0  B1 00017       CMPW    4(R0), #65535                                  ; 6814
                                        32  1E 0001D       BGEQU   2$
                          59            67  3C 0001F       MOVZWL  (R7), R9                                       ; 6821
                        0E  A0          59  B0 00022       MOVW    R9, 14(R0)
                          58        02  A7  3C 00026       MOVZWL  2(R7), R8                                      ; 6822
                        10  A0          58  B0 0002A       MOVW    R8, 16(R0)
                        12  A0      04  A7  B0 0002E       MOVW    4(R7), 18(R0)                                  ; 6823
                          51        08  AC  D0 00033       MOVL    OLD_FILE_ID, R1                                ; 6824
                        04  A7      04  A1  91 00037       CMPB    4(R1), 4(R7)
                                        28  12 0003C       BNEQ    4$
                                    12  A0  94 0003E       CLRB    18(R0)                                         ; 6825
                                        23  11 00041       BRB     4$                                             ; 6808
                          51        01  A0  9A 00043 1$:    MOVZBL  1(R0), R1                                      ; 6833
                          51          6041  3E 00047       MOVAW   (R0)[R1], MAP_AREA
                      FF  8F              61  91 0004B       CMPB    (MAP_AREA), #255                               ; 6838
                                        06  1F 0004F       BLSSU   3$
                          50      08C8  8F  3C 00051 2$:    MOVZWL  #2248, R0                                      ; 6840
                                        04  00056          RET
                          59            67  3C 00057 3$:    MOVZWL  (R7), R9                                       ; 6845
                        02  A1          59  B0 0005A       MOVW    R9, 2(MAP_AREA)
                          58        02  A7  3C 0005E       MOVZWL  2(R7), R8                                      ; 6846
                        04  A1          58  B0 00062       MOVW    R8, 4(MAP_AREA)
                          56        0C  AC  D0 00066 4$:    MOVL    NEW_HDR, R6                                    ; 6854
                          56            50  D1 0006A       CMPL    R0, R6
                                        1D  12 0006D       BNEQ    6$
                          0B        02  AA  E9 0006F       BLBC    QUAL+14, 5$                                    ; 6861
                          51        08  AC  D0 00073       MOVL    OLD_FILE_ID, R1                                ; 6862
                        04  A1        43  AA  91 00077       CMPB    QUAL+79, 4(R1)
                                        14  12 0007C       BNEQ    7$
                                        50  DD 0007E 5$:    PUSHL   R0                                             ; 6865
                                    08  AC  DD 00080       PUSHL   OLD_FILE_ID
                      D5E6  CF              02  FB 00083       CALLS   #2, WRITE_HEADER
                          07            50  E8 00088       BLBS    STATUS, 7$                                     ; 6866
                                        04  0008B          RET
                66            60    0200  8F  28 0008C 6$:    MOVC3   #512, (R0), (R6)                           ; 6870
                          51      0E  A6  9E 00092 7$:    MOVAB   14(R6), R1                                     ; 6897
                          02      07  A6  91 00096       CMPB    7(R6), #2                                      ; 6873
```

```
                                    65  12 0009A          BNEQ    9$                                          ;
                              08    59  B0 0009C          MOVW    R9, 8(R6)                                   ; 6879
                              0A    58  B0 000A0          MOVW    R8, 10(R6)                                  ; 6880
                                 0C A6  94 000A4          CLRB    12(R6)                                      ; 6881
                              0D    05  A7 90 000A7       MOVB    5(R7), 13(R6)                               ; 6882
                           50    066C  CA D0 000AC       MOVL    CURRENT_MTL, R0                             ; 6887
                              42    18  A0 D0 000B1       MOVL    24(R0), 66(R6)                              ;
                              46    1C  A0 B0 000B6       MOVW    28(R0), 70(R6)                              ; 6889
                              04    46  A6 91 000BB       CMPB    70(R6), 4(R7)                               ; 6890
                                    03  12 000C0          BNEQ    8$                                          ;
                                 46 A6  94 000C2          CLRB    70(R6)                                      ; 6891
                                 04 A6  B6 000C5 8$:      INCW    4(R6)                                       ; 6896
                                    61  B4 000C8          CLRW    (R1)                                        ; 6897
                                 10 A6  D4 000CA          CLRL    16(R6)                                      ; 6898
                                 3A A6  94 000CD          CLRB    58(R6)                                      ; 6900
                              34    C0  8F 8A 000D0       BICB2   #192, 52(R6)                                ; 6902
                                 18 A6  7C 000D5          CLRQ    24(R6)                                      ; 6903
                                 20 A6  B4 000D8          CLRW    32(R6)                                      ; 6905
                        01 A6       66  0A 81 000DB       ADDB3   #10, (R6), 1(R6)                            ; 6911
                              02    FFFF  8F B0 000E0       MOVW    #65535, 2(R6)                               ; 6912
                                    01 A6 9A 000E6          MOVZBL  1(R6), R0                                   ; 6914
                                    50  CE 000EA          MNEGL   R0, R1                                      ;
                                    02  C4 000ED          MULL2   #2, R1                                      ;
                                 0200   C1 9E 000F0       MOVAB   512(R1), R1                                 ;
                                     6640  3E 000F5       MOVAW   (R6)[R0], R7                                ;
                     51       00        6E  00  2C 000F9  MOVC5   #0, (SP), #0, R1, (R7)                      ;
                                        67  000FE                                                             ;
                                    2C  11 000FF          BRB     10$                                         ; 6873
                                 01 A6  9A 00101 9$:      MOVZBL  1(R6), R0                                   ; 6922
                                     6640  3E 00105       MOVAW   (R6)[R0], MAP_AREA                          ;
                              02    59  B0 00109          MOVW    R9, 2(R                                     ; 6927
                              04    58  B0 0010D          MOVW    R8, 4(                                      ; 6928
                                 04 A1  7C 00111          CLRQ    4(R1)                                       ; 6933
                                 0C A1  B4 00114          CLRW    12(R1)                                      ; 6935
                                    60  96 00117          INCB    (MAP_AREA)                                  ; 6936
                                 02 A0  D4 00119          CLRL    2(MAP_AREA)                                 ; 6937
                                 08 A0  94 0011C          CLRB    8(MAP_AREA)                                 ; 6939
                                    09  A0 9A 0011F       MOVZBL  9(MAP_AREA), R1                             ; 6940
                                    51  02  C4 00123       MULL2   #2, R1                                      ;
                     51       00        6E  00  2C 00126  MOVC5   #0, (SP), #0, R1, 10(MAP_AREA)              ;
                                        0A  A0  0012B                                                         ;
                                    50  01  D0 0012D 10$:  MOVL    #1, R0                                      ; 6947
                                        04  00130         RET                                                 ;

; Routine Size:  305 bytes,    Routine Base:  CODE + 2CD2
```

```
;  5435     6948  1 %SBTTL 'STA_CREATE - create QIO service routine'
;  5436     6949  1 ROUTINE STA_CREATE (EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)=
;  5437     6950  1
;  5438     6951  1 !++
;  5439     6952  1 !
;  5440     6953  1 ! FUNCTIONAL DESCRIPTION:
;  5441     6954  1 !       This routine executes IO$_CREATE in the standalone environment.
;  5442     6955  1 !
;  5443     6956  1 ! INPUT PARAMETERS:
;  5444     6957  1 !       As for $QIO(W) system service.  However, a nonzero P6 points to
;  5445     6958  1 !       OUTPUT_ATTBUF, which indicates that the IO$_CREATE refers to a file
;  5446     6959  1 !       on an image output volume.
;  5447     6960  1 !
;  5448     6961  1 ! IMPLICIT INPUTS:
;  5449     6962  1 !       CURRENT_MTL     - Pointer to MTL for selected volume set.
;  5450     6963  1 !
;  5451     6964  1 ! OUTPUT PARAMETERS:
;  5452     6965  1 !       NONE
;  5453     6966  1 !
;  5454     6967  1 ! IMPLICIT OUTPUTS:
;  5455     6968  1 !       NONE
;  5456     6969  1 !
;  5457     6970  1 ! ROUTINE VALUE:
;  5458     6971  1 !       Completion status.
;  5459     6972  1 !
;  5460     6973  1 ! SIDE EFFECTS:
;  5461     6974  1 !       NONE
;  5462     6975  1 !
;  5463     6976  1 !--
;  5464     6977  1
;  5465     6978  2 BEGIN
;  5466     6979  2 MAP
;  5467     6980  2         FUNC:           BBLOCK,         ! I/O function code
;  5468     6981  2         P1:             REF BBLOCK,     ! Descriptor for FIB
;  5469     6982  2         P2:             REF BBLOCK,     ! Descriptor for filename
;  5470     6983  2         P6:             REF BBLOCK;     ! Pointer to OUTPUT_ATTBUF
;  5471     6984  2 LOCAL
;  5472     6985  2         FIB:            REF BBLOCK,     ! Pointer to FIB
;  5473     6986  2         HEADER:         REF BBLOCK,     ! Pointer to primary header
;  5474     6987  2         EXT_HDR:        BBLOCK[512],    ! Buffer for extension header
;  5475     6988  2         CUR_HDR:        REF BBLOCK,     ! Pointer to current header
;  5476     6989  2         CUR_FID:        REF BBLOCK,     ! Pointer to current file ID
;  5477     6990  2         CHANNEL,                        ! Channel assigned to RVN
;  5478     6991  2         CRT:            REF BBLOCK,     ! Pointer to CRT entry
;  5479     6992  2         STATUS,                         ! Status return
;  5480     6993  2         TCOUNT,                         ! Total allocated block count
;  5481     6994  2         HCOUNT;                         ! Count of file headers
;  5482     6995  2
;  5483     6996  2
;  5484     6997  2 ! Initialize and check parameters.
;  5485     6998  2 !
;  5486     6999  2 IF .CURRENT_MTL[MTL_WINDOW] NEQ 0 THEN RETURN SS$_FILALRACC;
;  5487     7000  2 QUEUE_HEADERS[0] = QUEUE_HEADERS[1] = QUEUE_HEADERS[0];
;  5488     7001  2 QUEUE_HEADERS[2] = QUEUE_HEADERS[3] = QUEUE_HEADERS[2];
;  5489     7002  2 QUEUE_HEADERS[4] = QUEUE_HEADERS[5] = QUEUE_HEADERS[4];
;  5490     7003  2
;  5491     7004  2
```

```
;  5492          7005  2 ! Get file ID.
;  5493          7006  2 !
;  5494          7007  2 FIB = .P1[DSC$A_POINTER];
;  5495          7008  2 CURRENT_MTL[MTL_FID_NUM] = .FIB[FIB$W_FID_NUM];
;  5496          7009  2 CURRENT_MTL[MTL_FID_SEQ] = .FIB[FIB$W_FID_SEQ];
;  5497          7010  2 CURRENT_MTL[MTL_FID_RVNW] = .FIB[FIB$Q_FID_RVN];
;  5498          7011  2 IF .FIB[FIB$L_EXVBN] EQL 0 THEN FIB[FIB$L_EXVBN] = 1;
;  5499          7012  2 CURRENT_MTL[MTL_ACLFL] = CURRENT_MTL[MTL_ACLFL];
;  5500          7013  2 CURRENT_MTL[MTL_ACLBL] = CURRENT_MTL[MTL_ACLFL];
;  5501          7014  2 CURRENT_MTL[MTL_NEW_ACL] = 0;
;  5502          7015  2
;  5503          7016  2 IF .FIB[FIB$W_FID_NUM] EQL 0
;  5504          7017  2 AND .FIB[FIB$Q_FID_RVN] EQL 0
;  5505          7018  2 THEN SIGNAL (BACKUP$_INVFID, 0);
;  5506          7019  2
;  5507          7020  2 ! Create header.
;  5508          7021  2 !
;  5509          7022  2 HEADER = .CURRENT_MTL[MTL_HEADER];
;  5510          7023  2 CH$FILL(0, 512, .HEADER);
;  5511          7024  2 IF .CURRENT_MTL[MTL_STRUCLEV] EQL 1
;  5512          7025  2 THEN
;  5513          7026  3     BEGIN
;  5514          7027  3     LOCAL
;  5515          7028  3         NAMEBLOCK:      BBLOCK[NMB$C_LENGTH];
;  5516          7029  3
;  5517          7030  3     HEADER[FH1$B_IDOFFSET] = FH1$C_LENGTH/2;
;  5518          7031  3     HEADER[FH1$B_MPOFFSET] = (FH1$C_LENGTH+FI1$C_LENGTH)/2;
;  5519          7032  3     HEADER[FH1$W_FID_NUM] = .FIB[FIB$W_FID_NUM];
;  5520          7033  3     HEADER[FH1$W_FID_SEQ] = .FIB[FIB$W_FID_SEQ];
;  5521          7034  3     HEADER[FH1$W_STRUCLEV] = FH1$C_LEVEL1;
;  5522          7035  3     IF .FIB[FIB$V_ALCON] THEN HEADER[FH1$V_CONTIG] = TRUE;
;  5523          7036  3     MAKE_NAMEBLOCK(.P2[DSC$W_LENGTH], .P2[DSC$A_POINTER], NAMEBLOCK);
;  5524          7037  3     CH$MOVE(
;  5525          7038  3         10,
;  5526          7039  3         NAMEBLOCK[NMB$W_NAME],
;  5527          7040  3         BBLOCK[.HEADER + FH1$C_LENGTH, FI1$W_FILENAME]);
;  5528          7041  3     BBLOCK[.HEADER + FH1$C_LENGTH + FI1$C_LENGTH, FM1$B_COUNTSIZE] = 1;
;  5529          7042  3     BBLOCK[.HEADER + FH1$C_LENGTH + FI1$C_LENGTH, FM1$B_LBNSIZE] = 3;
;  5530          7043  3     BBLOCK[.HEADER + FH1$C_LENGTH + FI1$C_LENGTH, FM1$B_AVAIL] =
;  5531          7044  3         (512-2-FH1$C_LENGTH-FI1$C_LENGTH-FM1$C_LENGTH)/2;
;  5532          7045  3     END
;  5533          7046  2 ELSE
;  5534          7047  3     BEGIN
;  5535          7048  3     HEADER[FH2$B_IDOFFSET] = FH2$C_LENGTH/2;
;  5536          7049  3     HEADER[FH2$B_MPOFFSET] = (FH2$C_LENGTH+FI2$C_LENGTH)/2;
;  5537          7050  3     HEADER[FH2$B_ACOFFSET] = $BYTEOFFSET(FH2$W_CHECKSUM)/2;
;  5538          7051  3     HEADER[FH2$B_RSOFFSET] = $BYTEOFFSET(FH2$W_CHECKSUM)/2;
;  5539          7052  3     HEADER[FH2$B_STRUCVER] = 1;
;  5540          7053  3     HEADER[FH2$B_STRUCLEV] = 2;
;  5541          7054  3     HEADER[FH2$W_FID_NUM] = .FIB[FIB$W_FID_NUM];
;  5542          7055  3     HEADER[FH2$B_FID_NMX] = .FIB[FIB$B_FID_NMX];
;  5543          7056  3     HEADER[FH2$W_FID_SEQ] = .FIB[FIB$W_FID_SEQ];
;  5544          7057  3     HEADER[FH2$W_BK_FIDNUM] = .FIB[FIB$W_DID_NUM];
;  5545          7058  3     HEADER[FH2$W_BK_FIDSEQ] = .FIB[FIB$W_DID_SEQ];
;  5546          7059  3     HEADER[FH2$W_BK_FIDRVN] = .FIB[FIB$W_DID_RVN];
;  5547          7060  3     IF .FIB[FIB$V_ALCON] THEN HEADER[FH2$V_CONTIG] = TRUE;
;  5548          7061  3     HEADER[FH2$L_HIGHWATER] = 1;
```

```
; 5549          7062  3     IF .P6 NEQ 0
; 5550          7063  3     THEN HEADER[FH2$L_HIGHWATER] = .P6[FAR_HIGHWATER];
; 5551          7064  3     CH$COPY(
; 5552          7065  3         .P2[DSC$W_LENGTH],
; 5553          7066  3         .P2[DSC$A_POINTER],
; 5554          7067  3         %C' ',
; 5555          7068  3         FI2$S_FILENAME,
; 5556          7069  3         BBLOCK[.HEADER + FH2$C_LENGTH, FI2$T_FILENAME]);
; 5557          7070  3     CH$COPY(
; 5558          7071  3         MAX (.P2[DSC$W_LENGTH]-FI2$S_FILENAME, 0),
; 5559          7072  3         .P2[DSC$A_POINTER]+FI2$S_FILENAME,
; 5560          7073  3         %C' ',
; 5561          7074  3         FI2$S_FILENAMEXT,
; 5562          7075  3         BBLOCK[.HEADER + FH2$C_LENGTH, FI2$T_FILENAMEXT]);
; 5563          7076  2     END;
; 5564          7077  2
; 5565          7078  2
; 5566          7079  2 ! Write attributes.
; 5567          7080  2 !
; 5568          7081  2 STATUS = WRITE_ATTRIBUTES(.HEADER, .P5, .FIB);
; 5569          7082  2 IF NOT .STATUS THEN RETURN .STATUS;
; 5570          7083  2
; 5571          7084  2
; 5572          7085  2 ! Create window.
; 5573          7086  2 !
; 5574          7087  2 STATUS = CREATE_WINDOW(
; 5575          7088  2     .HEADER, 0, CURRENT_MTL[MTL_WINDOW],
; 5576          7089  2     .FIB[FIB$L_EXVBN], MAXU(10, .FIB[FIB$B_WSIZE]));
; 5577          7090  2 IF NOT .STATUS THEN RETURN .STATUS;
; 5578          7091  2 BBLOCK [.CURRENT_MTL[MTL_WINDOW], WCB_CUR_HWM] = .FIB[FIB$L_EXVBN];
; 5579          7092  2
; 5580          7093  2
; 5581          7094  2 ! Process placement data.
; 5582          7095  2 !
; 5583          7096  2 TCOUNT = 0;
; 5584          7097  2 HCOUNT = 1;
; 5585          7098  2 IF
; 5586          7099  2     BEGIN
; 5587          7100  2     IF .P6 NEQ 0
; 5588          7101  2         THEN .BBLOCK[P6[FAR_PLACEMENT], DSC$W_LENGTH] NEQ 0
; 5589          7102  2         ELSE FALSE
; 5590          7103  2     END
; 5591          7104  2 THEN
; 5592          7105  2     BEGIN
; 5593          7106  2     LOCAL
; 5594          7107  2         CRT:            REF BBLOCK,     ! Pointer to create list entry
; 5595          7108  2         FID:            REF BBLOCK,     ! Pointer to FID list entry
; 5596          7109  2         PLC:            REF BBLOCK,     ! Pointer to placement data
; 5597          7110  2         PLC_END:        REF BBLOCK;     ! Pointer beyond placement data
; 5598          7111  2
; 5599          7112  2
; 5600          7113  2     ! Placement data exists.  Allocate a CRT entry and FID entry for the file
; 5601          7114  2     ! ID in the FIB.
; 5602          7115  2     !
; 5603          7116  2     CRT = GET_VM(CRT_S_BLOCKS);
; 5604          7117  2     INSQUE(.CRT, .QUEUE_HEADERS[2]);
; 5605          7118  3     CRT[CRT_FID_FQHDR] = CRT[CRT_FID_BQHDR] = CRT[CRT_FID_FQHDR];
```

```
; 5606          7119  3     CRT[CRT_BLOCKS] = 0;
; 5607          7120  3     FID = GET_VM(CRT_S_FID);
; 5608          7121  3     INSQUE(.FID, .CRT[CRT_FID_FQHDR]);
; 5609          7122  3     FID[CRT_FID_NUM] = .FIB[FIB$W_FID_NUM];
; 5610          7123  3     FID[CRT_FID_SEQ] = .FIB[FIB$W_FID_SEQ];
; 5611          7124  3     FID[CRT_FID_RVNW] = .FIB[FIB$B_FID_RVN];
; 5612          7125  3     PLC = .BBLOCK[P6[FAR_PLACEMENT], DSC$A_POINTER];
; 5613          7126  3     PLC_END = .PLC + .BBLOCK[P6[FAR_PLACEMENT], DSC$W_LENGTH];
; 5614          7127  3     WHILE .PLC LSSA .PLC_END DO
; 5615          7128  4         BEGIN
; 5616          7129  4         LOCAL
; 5617          7130  4             TYPE;
; 5618          7131  4
; 5619          7132  4         TYPE = .(.PLC)<0,8>;
; 5620          7133  4         PLC = .PLC + 1;
; 5621          7134  4         CASE .TYPE FROM BSA$K_PLC_FID TO BSA$K_PLC_PLLBN OF
; 5622          7135  4             SET
; 5623          7136  4
; 5624          7137  4             [BSA$K_PLC_FID]:
; 5625          7138  5                 BEGIN
; 5626          7139  5                 IF
; 5627          7140  6                     BEGIN
; 5628          7141  6                     IF .QUAL[QUAL_OF11] AND .QUAL[QUAL_VOLU]
; 5629          7142  6                     THEN
; 5630          7143  6                         TRUE
; 5631          7144  6                     ELSE
; 5632          7145  7                         BEGIN
; 5633          7146  7                         CRT = .QUEUE_HEADERS[2];
; 5634          7147  7                         WHILE .CRT NEQA QUEUE_HEADERS[2] DO
; 5635          7148  8                             BEGIN
; 5636          7149  8                             IF .PLC[FID$B_RVN] EQL .BBLOCK[.CRT[CRT_FID_FQHDR], CRT_FID_RVN] THEN EXITLOOP F
; 5637          7150  8                             CRT = .CRT[CRT_FLINK];
; 5638          7151  8                             END
; 5639          7152  7                         END
; 5640          7153  6                     END
; 5641          7154  5                 THEN
; 5642          7155  6                     BEGIN
; 5643          7156  6                     CRT = GET_VM(CRT_S_BLOCKS);
; 5644          7157  6                     INSQUE(.CRT, .QUEUE_HEADERS[3]);
; 5645          7158  6                     CRT[CRT_FID_FQHDR] = CRT[CRT_FID_BQHDR] = CRT[CRT_FID_FQHDR];
; 5646          7159  6                     CRT[CRT_BLOCKS] = 0;
; 5647          7160  5                     END;
; 5648          7161  5                 FID = GET_VM(CRT_S_FID);
; 5649          7162  5                 INSQUE(.FID, .CRT[CRT_FID_BQHDR]);
; 5650          7163  5                 FID[CRT_FID_NUM] = .PLC[FID$W_NUM];
; 5651          7164  5                 FID[CRT_FID_SEQ] = .PLC[FID$W_SEQ];
; 5652          7165  5                 FID[CRT_FID_RVNW] = .PLC[FID$B_RVN];
; 5653          7166  5                 PLC = .PLC + BSA$S_PLC_FID;
; 5654          7167  4                 END;
; 5655          7168  4
; 5656          7169  4             [BSA$K_PLC_COUNT]:
; 5657          7170  5                 BEGIN
; 5658          7171  5                 CRT[CRT_BLOCKS] = .CRT[CRT_BLOCKS] + ..PLC;
; 5659          7172  5                 PLC = .PLC + BSA$S_PLC_COUNT;
; 5660          7173  4                 END;
; 5661          7174  4
; 5662          7175  4             [BSA$K_PLC_PLACE]:
```

```
; 5663          7176  5                 BEGIN
; 5664          7177  5                 CRT[CRT_BLOCKS] = .CRT[CRT_BLOCKS] + .PLC[BSA$L_PLC_COUNT];
; 5665          7178  5                 PLC = .PLC + BSA$S_PLC_PLACE;
; 5666          7179  4                 END;
; 5667          7180  4
; 5668          7181  4             [BSA$K_PLC_PLLBN]:
; 5669          7182  5                 BEGIN
; 5670          7183  5                 CRT[CRT_BLOCKS] = .CRT[CRT_BLOCKS] + .PLC[BSA$L_PLC_COUNT];
; 5671          7184  5                 PLC = .PLC + BSA$S_PLC_PLLBN;
; 5672          7185  4                 END;
; 5673          7186  4
; 5674          7187  4             [OUTRANGE]:
; 5675          7188  4                 RETURN CREATE_CLEANUP(BACKUP$_INVATTVAL);
; 5676          7189  4
; 5677          7190  4             TES;
; 5678          7191  3         END;
; 5679          7192  3     END
; 5680          7193  2 ELSE
; 5681          7194  3     BEGIN
; 5682          7195  3     LOCAL
; 5683          7196  3         CRT:            REF BBLOCK,     ! Pointer to create list entry
; 5684          7197  3         FID:            REF BBLOCK;     ! Pointer to FID list entry
; 5685          7198  3
; 5686          7199  3
; 5687          7200  3     ! No placement data exists.  Allocate a degenerate list containing just
; 5688          7201  3     ! the file ID in the FIB and BLOCKS equal to EXSZ.
; 5689          7202  3     !
; 5690          7203  3     CRT = GET_VM(CRT_S_BLOCKS);
; 5691          7204  3     INSQUE(.CRT, .QUEUE_HEADERS[2]);
; 5692          7205  3     CRT[CRT_FID_FQHDR] = CRT[CRT_FID_BQHDR] = CRT[CRT_FID_FQHDR];
; 5693          7206  3     CRT[CRT_BLOCKS] = .FIB[FIB$L_EXSZ];
; 5694          7207  3     FID = GET_VM(CRT_S_FID);
; 5695          7208  3     INSQUE(.FID, .CRT[CRT_FID_FQHDR]);
; 5696          7209  3     FID[CRT_FID_NUM] = .FIB[FIB$W_FID_NUM];
; 5697          7210  3     FID[CRT_FID_SEQ] = .FIB[FIB$W_FID_SEQ];
; 5698          7211  3     FID[CRT_FID_RVNW] = .FIB[FIB$B_FID_RVN];
; 5699          7212  2     END;
; 5700          7213  2
; 5701          7214  2
; 5702          7215  2 ! Allocate space, append map pointers and window pointers.
; 5703          7216  2 !
; 5704          7217  2 CRT = .QUEUE_HEADERS[2];
; 5705          7218  2 CUR_HDR = .HEADER;
; 5706          7219  2 CUR_FID = FIB[FIB$W_FID];
; 5707          7220  2 IF .FIB[FIB$L_EXSZ] NEQ 0
; 5708          7221  2 THEN
; 5709          7222  3     BEGIN
; 5710          7223  3     WHILE .TCOUNT LSSU .FIB[FIB$L_EXSZ] DO
; 5711          7224  4         BEGIN
; 5712          7225  4         LOCAL
; 5713          7226  4             FID:                REF BBLOCK,     ! Pointer to FID entry
; 5714          7227  4             RCOUNT;                             ! Requested allocation from this CRT
; 5715          7228  4
; 5716          7229  4
; 5717          7230  4         REMQUE(.CRT[CRT_FID_FQHDR], FID);       ! Get first FID entry
; 5718          7231  4         INSQUE(.FID, QUEUE_HEADERS[4]);
; 5719          7232  4
```

```
;  5720          7233  4
;  5721          7234  4          ! Generate an extension header if this is not the primary file ID.
;  5722          7235  4          !
;  5723          7236  4          IF .CRT NEQA .QUEUE_HEADERS[2]
;  5724          7237  4          THEN
;  5725          7238  5              BEGIN
;  5726          7239  5              STATUS = CREATE_EXTHDR(.CUR_HDR, .CUR_FID, EXT_HDR, FID[CRT_FID]);
;  5727          7240  5              IF NOT .STATUS THEN RETURN CREATE_CLEANUP(.STATUS);
;  5728          7241  5              CUR_HDR = EXT_HDR;
;  5729          7242  5              CUR_FID = FID[CRT_FID];
;  5730          7243  5              HCOUNT = .HCOUNT + 1;
;  5731          7244  4              END;
;  5732          7245  4
;  5733          7246  4
;  5734          7247  4          RCOUNT = MINU(.FIB[FIB$L_EXSZ] - . TCOUNT, .CRT[CRT_BLOCKS]);
;  5735          7248  4          IF
;  5736          7249  4              NOT .QUAL[QUAL_OF11] OR      ! Sequential disk output
;  5737          7250  4              NOT .QUAL[QUAL_VOLU] OR      ! Not /VOLUME restore
;  5738          7251  4              .QUAL[QUAL_VOLU_VALUE] EQL .FID[CRT_FID_RVN]
;  5739          7252  4          THEN
;  5740          7253  5              BEGIN
;  5741          7254  5              LOCAL
;  5742          7255  5                  VCB:            REF BBLOCK;     ! VCB for RVN
;  5743          7256  5
;  5744          7257  5
;  5745          7258  5              IF .FID[CRT_FID_RVN] - .CURRENT_MTL[MTL_RVN_BASE] GEQU .CURRENT_MTL[MTL_SETCOUNT]
;  5746          7259  5              THEN
;  5747          7260  5                  RETURN CREATE_CLEANUP(SS$_DEVNOTMOUNT);
;  5748          7261  5
;  5749          7262  5              CURRENT_VCB = VCB = .CURRENT_MTL[MTL_VCB(.FID[CRT_FID_RVN]-.CURRENT_MTL[MTL_RVN_BASE])];
;  5750          7263  5
;  5751          7264  5              IF NOT .VCB[VCB_INIT_DONE]
;  5752          7265  5              THEN
;  5753          7266  5                  SIGNAL(BACKUP$_NOVOLDATA, 1, VCB[VCB_DEVICE]);
;  5754          7267  5
;  5755          7268  5
;  5756          7269  5              WHILE TRUE DO
;  5757          7270  6                  BEGIN
;  5758          7271  6                  LOCAL
;  5759          7272  6                      EXT:        REF BBLOCK;     ! Pointer to extent list entry
;  5760          7273  6
;  5761          7274  6
;  5762          7275  6                  ! Generate extent list entry for new extent.
;  5763          7276  6                  !
;  5764          7277  6                  EXT = GET_VM(EXT_S_ENTRY);
;  5765          7278  6                  INSQUE(.EXT, .QUEUE_HEADERS[0]);
;  5766          7279  6                  EXT[EXT_VCB] = .VCB;
;  5767          7280  6
;  5768          7281  6
;  5769          7282  6                  ! Try to allocate remaining blocks.
;  5770          7283  6                  !
;  5771          7284  6                  STA_ALLOC_BEST(.RCOUNT, EXT[EXT_COUNT], EXT[EXT_LBN]);
;  5772          7285  6
;  5773          7286  6
;  5774          7287  6                  ! If no blocks were allocated, or if not enough blocks were
;  5775          7288  6                  ! allocated and the allocation is required to be contiguous,
;  5776          7289  6                  ! return device-full.
```

```
;  5777          7290  6                !
;  5778          7291  6                IF
;  5779          7292  6                    .EXT[EXT_COUNT] EQL 0 OR
;  5780          7293  6                    .EXT[EXT_COUNT] LSSU .RCOUNT AND .FIB[FIB$V_ALCON]
;  5781          7294  6                THEN
;  5782          7295  6                    RETURN CREATE_CLEANUP(SS$_DEVICEFULL);
;  5783          7296  6
;  5784          7297  6
;  5785          7298  6                ! Append the map pointer.  For ODS-1, append one maximal pointer at
;  5786          7299  6                ! a time so that header overflow can be cleanly detected.  If the
;  5787          7300  6                ! header should fill, allocate an extension header from the list
;  5788          7301  6                ! hanging from the CRT entry.  If all headers have been used, fail.
;  5789          7302  6                !
;  5790          7303  7                BEGIN LOCAL L;
;  5791          7304  7                L = .EXT[EXT_LBN];
;  5792          7305  7                WHILE .L LSSU .EXT[EXT_LBN] + .EXT[EXT_COUNT] DO
;  5793          7306  8                    BEGIN LOCAL C;
;  5794          7307  8                    C = .EXT[EXT_LBN] + .EXT[EXT_COUNT] - .L;
;  5795          7308  8                    IF NOT .VCB[VCB_ODS_2]
;  5796          7309  8                    THEN
;  5797          7310  9                        BEGIN
;  5798          7311  9                        IF .C GTRU 256 THEN C = 256;
;  5799          7312  9                        STATUS = MAKE_POINTER1(.CUR_HDR, .C, .L);
;  5800          7313  9                        END
;  5801          7314  8                    ELSE
;  5802          7315  8                        STATUS = MAKE_POINTER(.CUR_HDR, .C, .L);
;  5803          7316  8
;  5804          7317  8                    IF .STATUS
;  5805          7318  8                    THEN
;  5806          7319  8                        L = .L + .C
;  5807          7320  8                    ELSE IF REMQUE(.CRT[CRT_FID_FQHDR], FID)
;  5808          7321  8                    THEN
;  5809          7322  8                        RETURN CREATE_CLEANUP(SS$_HEADERFULL)
;  5810          7323  8                    ELSE
;  5811          7324  9                        BEGIN
;  5812          7325  9                        INSQUE(.FID, QUEUE_HEADERS[4]);
;  5813          7326  9                        STATUS = CREATE_EXTHDR(
;  5814          7327  9                            .CUR_HDR, .CUR_FID, EXT_HDR, FID[CRT_FID]);
;  5815          7328  9                        IF NOT .STATUS
;  5816          7329  9                            THEN RETURN CREATE_CLEANUP(.STATUS);
;  5817          7330  9                        CUR_HDR = EXT_HDR;
;  5818          7331  9                        CUR_FID = FID[CRT_FID];
;  5819          7332  9                        HCOUNT = .HCOUNT + 1;
;  5820          7333  8                        END;
;  5821          7334  7                    END;
;  5822          7335  6                END;
;  5823          7336  6
;  5824          7337  6
;  5825          7338  6                ! Append the window pointer.
;  5826          7339  6                !
;  5827          7340  6                ADD_WINDOW_MAP(
;  5828          7341  6                    .CURRENT_MTL[MTL_WINDOW],
;  5829          7342  6                    .FID[CRT_FID_RVN], .EXT[EXT_COUNT], .EXT[EXT_LBN]);
;  5830          7343  6
;  5831          7344  6
;  5832          7345  6                ! Count the allocation into the total, and determine if we need
;  5833          7346  6                ! to go around again for another extent.
```

```
;  5834     7347  6                    !
;  5835     7348  6                    TCOUNT = .TCOUNT + .EXT[EXT_COUNT];
;  5836     7349  6                    IF .EXT[EXT_COUNT] GEQU .RCOUNT THEN EXITLOOP;
;  5837     7350  6                    RCOUNT = .RCOUNT - .EXT[EXT_COUNT];
;  5838     7351  5                    END;
;  5839     7352  5                END
;  5840     7353  4            ELSE
;  5841     7354  5                BEGIN
;  5842     7355  5                ADD_BLACKHOLE_MAP(.CURRENT_MTL[MTL_WINDOW], .CRT[CRT_BLOCKS]);
;  5843     7356  5                TCOUNT = .TCOUNT + .CRT[CRT_BLOCKS];
;  5844     7357  4                END;
;  5845     7358  4            CRT = .CRT[CRT_FLINK];
;  5846     7359  3            END;
;  5847     7360  3        END
;  5848     7361  2    ELSE
;  5849     7362  3        BEGIN
;  5850     7363  3
;  5851     7364  3        ! If there are no blocks allocated to the file, it is still necessary
;  5852     7365  3        ! to take off the first entry of the FID queue.  This is necessary to
;  5853     7366  3        ! make the ACL building logic in STA_DEACCESS work correctly.
;  5854     7367  3        !
;  5855     7368  3        LOCAL
;  5856     7369  3            FID:            REF BBLOCK;     ! Pointer to FID entry
;  5857     7370  3
;  5858     7371  3
;  5859     7372  3        REMQUE(.CRT[CRT_FID_FQHDR], FID);   ! Get first FID entry
;  5860     7373  3        INSQUE(.FID, QUEUE_READERS[4]);
;  5861     7374  2        END;
;  5862     7375  2
;  5863     7376  2
;  5864     7377  2    ! Initialize HIBLK.
;  5865     7378  2    !
;  5866     7379  2    IF
;  5867     7380  2        NOT .QUAL[QUAL_OF11] OR     ! Sequential disk output
;  5868     7381  2        NOT .QUAL[QUAL_VOLU]        ! Not /VOLUME restore
;  5869     7382  2    THEN
;  5870     7383  2        IF .HEADER[FH2$B_STRUCLEV] EQL 1
;  5871     7384  2            THEN BBLOCK[HEADER[FH1$W_RECATTR], FAT$L_HIBLK] = ROT(.TCOUNT, 16)
;  5872     7385  2            ELSE BBLOCK[HEADER[FH2$W_RECATTR], FAT$L_HIBLK] = ROT(.TCOUNT, 16);
;  5873     7386  2
;  5874     7387  2
;  5875     7388  2    ! Write last extension header if it exists.
;  5876     7389  2    !
;  5877     7390  2    IF
;  5878     7391  2        .CUR_HDR EQLA EXT_HDR AND
;  5879     7392  3        (NOT .QUAL[QUAL_VOLU] OR .QUAL[QUAL_VOLU_VALUE] EQL .CUR_FID[FID$B_RVN])
;  5880     7393  2    THEN
;  5881     7394  3        BEGIN
;  5882     7395  3        STATUS = WRITE_HEADER(.CUR_FID, .CUR_HDR);
;  5883     7396  3        IF NOT .STATUS THEN RETURN CREATE_CLEANUP(.STATUS);
;  5884     7397  2        END;
;  5885     7398  2
;  5886     7399  2
;  5887     7400  2    ! Write header and set index file bitmap bit.
;  5888     7401  2    !
;  5889     7402  2    IF
;  5890     7403  2        NOT .QUAL[QUAL_OF11] OR     ! Sequential disk output
```

```
; 5891          7404  2     NOT .QUAL[QUAL_VOLU] OR     ! Not /VOLUME restore
; 5892          7405  2     .QUAL[QUAL_VOLO_VALUE] EQL .CURRENT_MTL[MTL_FID_RVN]
; 5893          7406  2 THEN
; 5894          7407  3     BEGIN
; 5895          7408  3     STATUS = WRITE_HEADER(CURRENT_MTL[MTL_FID], .HEADER);
; 5896          7409  3     IF NOT .STATUS THEN RETURN CREATE_CLEANUP(.STATUS);
; 5897          7410  2     END;
; 5898          7411  2
; 5899          7412  2
; 5900          7413  2 ! Quota table maintenance.
; 5901          7414  2 !
; 5902          7415  2 IF .HEADER[FH2$B_STRUCLEV] EQL 2 AND .QUAL[QUAL_OF11] AND NOT .QUAL[QUAL_VOLU]
; 5903          7416  2 THEN
; 5904          7417  3     BEGIN
; 5905          7418  3
; 5906          7419  3     ! Locate and record file ID of QUOTA.SYS.
; 5907          7420  3     !
; 5908          7421  3     IF
; 5909          7422  3         .BBLOCK[HEADER[FH2$W_RECATTR], FAT$B_RTYPE] EQL FAT$C_FIXED AND
; 5910          7423  3         .BBLOCK[HEADER[FH2$W_RECATTR], FAT$B_RATTRIB] EQL 0 AND
; 5911          7424  3         .BBLOCK[HEADER[FH2$W_RECATTR], FAT$W_RSIZE] EQL DQF$C_LENGTH AND
; 5912          7425  3         .HEADER[FH2$V_CONTIG] AND
; 5913          7426  3         .HEADER[FH2$W_BK_FIDNUM] EQL FID$C_MFD AND
; 5914          7427  3         .HEADER[FH2$B_BK_FIDNMX] EQL 0 AND
; 5915          7428  3         .HEADER[FH2$W_BK_FIDSEQ] EQL FID$C_MFD AND
; 5916          7429  3         .FIB[FIB$B_FID_RVN] EQL 1 AND
; 5917          7430  3         .P2[DSC$W_LENGTH] EQL %CHARCOUNT('QUOTA.SYS;1') AND
; 5918          7431  3         CH$EQL(
; 5919          7432  3             %CHARCOUNT('QUOTA.SYS;1'), .P2[DSC$A_POINTER],
; 5920          7433  3             %CHARCOUNT('QUOTA.SYS;1'), UPLIT BYTE('QUOTA.SYS;1'))
; 5921          7434  3     THEN
; 5922          7435  4         BEGIN
; 5923          7436  4         DQF_QUOTA_FID[FID$W_NUM] = .FIB[FIB$W_FID_NUM];
; 5924          7437  4         DQF_QUOTA_FID[FID$W_SEQ] = .FIB[FIB$W_FID_SEQ];
; 5925          7438  4         DQF_QUOTA_FID[FID$W_RVN] = .FIB[FIB$W_FID_RVN];
; 5926          7439  3         END;
; 5927          7440  3
; 5928          7441  3
; 5929          7442  3     ! Charge file space.
; 5930          7443  3     !
; 5931          7444  3     DQF_MODIFY_USAGE(.HEADER[FH2$L_FILEOWNER], .TCOUNT + .HCOUNT);
; 5932          7445  2     END;
; 5933          7446  2
; 5934          7447  2
; 5935          7448  2 ! Completed normally.
; 5936          7449  2 !
; 5937          7450  2 CREATE_CLEANUP(SS$_NORMAL)
; 5938          7451  1 END;


                31  3B  53  59  53  2E  41  54  4F  55  51  02E03 P.AAU:  .ASCII  \QUOTA.SYS;1\                          ;


                                                    0FFC 00000 STA_CREATE:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 6949
```

```
                                 5E      FDC8    CE  9E 00002          MOVAB   -568(SP), SP
                                 50 00000000'    EF  D0 00007          MOVL    CURRENT_MTL, R0                         ; 6999
                                          08     A0  D5 0000E          TSTL    8(R0)
                                                 05  13 00011          BEQL    1$
                                 50       A4     8F  9A 00013          MOVZBL  #164, R0
                                                 04  00017          RET
                                 51 00000000'    EF  9E 00018 1$:      MOVAB   QUEUE_HEADERS, R1                       ; 7000
                00000000'        EF              51  D0 0001F          MOVL    R1, QUEUE_HEADERS+4
                00000000'        EF              51  D0 00026          MOVL    R1, QUEUE_HEADERS
                                 51 00000000'    EF  9E 0002D          MOVAB   QUEUE_HEADERS+8, R1                     ; 7001
                00000000'        EF              51  D0 00034          MOVL    R1, QUEUE_HEADERS+12
                00000000'        EF              51  D0 0003B          MOVL    R1, QUEUE_HEADERS+8
                                 51 00000000'    EF  9E 00042          MOVAB   QUEUE_HEADERS+16, R1                    ; 7002
                00000000'        EF              51  D0 00049          MOVL    R1, QUEUE_HEADERS+20
                00000000'        EF              51  D0 00050          MOVL    R1, QUEUE_HEADERS+16
                                 51       1C     AC  D0 00057          MOVL    P1, R1                                  ; 7007
                                 56       04     A1  D0 0005B          MOVL    4(R1), FIB
                          18     A0       04     A6  D0 0005F          MOVL    4(FIB), 24(R0)                          ; 7008
                          1C     A0       08     A6  B0 00064          MOVW    8(FIB), 28(R0)                          ; 7010
                                          1C     A6  D5 00069          TSTL    28(FIB)                                 ; 7011
                                                 04  12 0006C          BNEQ    2$
                          1C     A6              01  D0 0006E          MOVL    #1, 28(FIB)
                          10     A0       10     A0  9E 00072 2$:      MOVAB   16(R0), 16(R0)                          ; 7012
                          14     A0       10     A0  9E 00077          MOVAB   16(R0), 20(R0)                          ; 7013
                          31     A0              02  8A 0007C          BICB2   #2, 49(R0)                              ; 7014
                                 5A       04     A6  3C 00080          MOVZWL  4(FIB), R10                             ; 7016
                                                 14  12 00084          BNEQ    3$
                                          08     A6  B5 00086          TSTW    8(FIB)                                  ; 7017
                                                 0F  12 00089          BNEQ    3$
                                                 7E  D4 0008B          CLRL    -(SP)                                   ; 7018
                                    00000000G    8F  DD 0008D          PUSHL   #BACKUP$_INVFID
                             00000000G   00      02  FB 00093          CALLS   #2, LIB$SIGNAL
                                 58 00000000'    EF  D0 0009A 3$:      MOVL    CURRENT_MTL, R8                         ; 7022
                                 57       0C     A8  D0 000A1          MOVL    12(R8), HEADER
          0200  8F       00                     6E  00 2C 000A5          MOVC5   #0, (SP), #0, #512, (HEADER)            ; 7023
                                                     67 000AC
                                 59       20     AC  D0 000AD          MOVL    P2, R9                                  ; 7036
                                 01       1E     A8  91 000B1          CMPB    30(R8), #1                              ; 7024
                                                 40  12 000B5          BNEQ    5$
                                 67     2E17     8F  B0 000B7          MOVW    #11799, (HEADER)                        ; 7030
                          02     A7              5A  B0 000BC          MOVW    R10, 2(HEADER)                          ; 7032
                          04     A7       06     A6  B0 000C0          MOVW    6(FIB), 4(HEADER)                       ; 7033
                          06     A7     0101     8F  B0 000C5          MOVW    #257, 6(HEADER)                         ; 7034
                                 05       16     A6  E9 000CB          BLBC    22(FIB), 4$                             ; 7035
                          0C     A7       80     8F  88 000CF          BISB2   #128, 12(HEADER)
                                          10     AE  9F 000D4 4$:      PUSHAB  NAMEBLOCK                               ; 7036
                                          04     A9  DD 000D7          PUSHL   4(R9)
                                 7E       20     BC  3C 000DA          MOVZWL  @P2, -(SP)
                             00000000G   00      03  FB 000DE          CALLS   #3, MAKE_NAMEBLOCK
                    2E   A7       16     AE  0A 28 000E5          MOVC3   #10, NAMEBLOCK+6, 46(HEADER)            ; 7040
                                 62     A7     0301     8F  B0 000EB          MOVW    #769, 98(HEADER)                        ; 7041
                                 65     A7              34  8E 000F1          MNEGB   #52, 101(HEADER)                        ; 7044
                                                        60  11 000F5          BRB     9$                                      ; 7024
                                                     67 FFFF6428    8F  D0 000F7 5$:      MOVL    #-39896, (HEADER)                       ; 7048
                                 06     A7     0201     8F  B0 000FE          MOVW    #513, 6(HEADER)                         ; 7052
                                 08     A7              5A  B0 00104          MOVW    R10, 8(HEADER)                          ; 7054
                                 0D     A7       09     A6  90 00108          MOVB    9(FIB), 13(HEADER)                      ; 7055
```

```
                                0A   A7      06   A6  B0 0010D          MOVW    6(FIB), 10(HEADER)                          ; 7056
                                42   A7      0A   A6  D0 00112          MOVL    10(FIB), 66(HEADER)                         ; 7057
                                46   A7      0E   A6  B0 00117          MOVW    14(FIB), 70(HEADER)                         ; 7059
                                     05      16   A6  E9 0011C          BLBC    22(FIB), 6$                                 ; 7060
                                34   A7      80   8F  88 00120          BISB2   #128, 52(HEADER)
                                4C   A7              01  D0 00125 6$:    MOVL    #1, 76(HEADER)                              ; 7061
                                     50      30   AC  D0 00129          MOVL    P6, R0                                      ; 7062
                                             06   13  13 0012D          BEQL    7$
                                4C   A7    0088   C0  D0 0012F          MOVL    136(R0), 76(HEADER)                         ; 7063
                                     58      04   A9  D0 00135 7$:     MOVL    4(R9), R8                                   ; 7066
          14            20           68      20   BC  2C 00139          MOVC5   @P2, (R8), #32, #20, 80(HEADER)             ; 7069
                                     50      A7         0013F
                                     50      20   BC  3C 00141          MOVZWL  @P2, R0                                     ; 7071
                                     50          14   C2 00145          SUBL2   #20, R0
                                                 02   18 00148          BGEQ    8$
                                                 50   D4 0014A          CLRL    R0
    0042  8F            20       14   A8              50  2C 0014C 8$:    MOVC5   R0, 20(R8), #32, #66, 134(HEADER)           ; 7075
                                           0086  C7         00154
                                                 56   DD 00157 9$:     PUSHL   FIB                                         ; 7081
                                           2C    AC   DD 00159          PUSHL   P5
                                                 57   DD 0015C          PUSHL   HEADER
                                0000V      CF             03   FB 0015E          CALLS   #3, WRITE_ATTRIBUTES
                                     6E              50   D0 00163          MOVL    R0, STATUS
                                     26              6E   E9 00166          BLBC    STATUS, 11$                                 ; 7082
                                     7E      03      A6   98 00169          CVTBL   3(FIB), -(SP)                               ; 7089
                                     0A              6E   91 0016D          CMPB    (SP), #10
                                                     03   1E 00170          BGEQU   10$
                                     6E              0A   D0 00172          MOVL    #10, (SP)
                                           1C        A6   DD 00175 10$:     PUSHL   28(FIB)
                                7E 00000000'   EF        08   C1 00178          ADDL3   #8, CURRENT_MTL, -(SP)              ; 7088
                                                 7E   D4 00180          CLRL    -(SP)
                                                 57   DD 00182          PUSHL   HEADER
                                D837   CF             05   FB 00184          CALLS   #5, CREATE_WINDOW
                                     6E              50   D0 00189          MOVL    R0, STATUS
                                     04              6E   E8 0018C          BLBS    STATUS, 12$                                 ; 7090
                                     50              6E   D0 0018F 11$:     MOVL    STATUS, R0
                                                     04   00192          RET
                                     50 00000000'    EF   D0 00193 12$:     MOVL    CURRENT_MTL, R0                         ; 7091
                                     50      08      A0   D0 0019A          MOVL    8(R0), R0
                                0C   A0      1C      A6   D0 0019E          MOVL    28(FIB), 12(R0)
                                                     54   D4 001A3          CLRL    TCOUNT                                  ; 7096
                                0C   AE              01   D0 001A5          MOVL    #1, HCOUNT                              ; 7097
                                     50      30      AC   D0 001A9          MOVL    P6, R0                                  ; 7100
                                                     03   13 001AD          BEQL    13$
                                             08      A0   B5 001AF          TSTW    8(R0)                                   ; 7101
                                                     03   12 001B2 13$:     BNEQ    14$
                                                 00F5     31 001B4          BRW     28$
                                                     14   DD 001B7 14$:     PUSHL   #20                                     ; 7116
                           00000000G     00             01   FB 001B9          CALLS   #1, GET_VM
                                     52              50   D0 001C0          MOVL    R0, CRT
                           00000000'     FF             62   0E 001C3          INSQUE  (CRT), @QUEUE_HEADERS+8                 ; 7117
                                     50      08      A2   9E 001CA          MOVAB   8(CRT), R0                              ; 7118
                                0C   A2              50   D0 001CE          MOVL    R0, 12(CRT)
                                08   A2              50   D0 001D2          MOVL    R0, 8(CRT)
                                             10      A2   D4 001D6          CLRL    16(CRT)                                 ; 7119
                                                     0E   DD 001D9          PUSHL   #14                                     ; 7120
                           00000000G     00             01   FB 001DB          CALLS   #1, GET_VM
```

```
                                55              50  DO 001E2          MOVL    R0, FID                                 ;
                             08 B2              65  0E 001E5          INSQUE  (FID), @8(CRT)                          ; 7121
                             08 A5          04  A6  D0 001E9          MOVL    4(FIB), 8(FID)                          ; 7122
                             0C A5          08  A6  B0 001EE          MOVW    8(FIB), 12(FID)                         ; 7124
                                50          30  AC  D0 001F3          MOVL    P6, R0                                  ; 7125
                                53          0C  A0  D0 001F7          MOVL    12(R0), PLC
                                58          08  A0  3C 001FB          MOVZWL  8(R0), PLC_END                          ; 7126
                                58              53  C0 001FF          ADDL2   PLC, PLC_END
                                58              53  D1 00202 15$:      CMPL    PLC, PLC_END                            ; 7127
                                                03  1F 00205          BLSSU   16$
                                              00DD  31 00207          BRW     29$
                                50              83  9A 0020A 16$:     MOVZBL  (PLC)+, TYPE                            ; 7132
                          03                    01  CF 0020D          CASEL   TYPE, #1, #3                            ; 7134
                0090            0086                0080    0011      00211 17$:  .WORD   18$-17$,-
                                                                                          23$-17$,-
                                                                                          24$-17$,-
                                                                                          26$-17$
                                00000000G   8F      DD 00219          PUSHL   #BACKUP$_INVATTVAL                      ; 7188
                                              03AA  31 0021F          BRW     65$
                          07 00000000'      EF      06  E1 00222 18$:  BBC     #6, QUAL+15, 19$                        ; 7141
                                23 00000000'  EF      E8 0022A          BLBS    QUAL+14, 21$
                                52 00000000'  EF      D0 00231 19$:     MOVL    QUEUE_HEADERS+8, CRT                    ; 7146
                                50 00000000'  EF      9E 00238 20$:     MOVAB   QUEUE_HEADERS+8, R0                     ; 7147
                                50              52  D1 0023F          CMPL    CRT, R0
                                                10  13 00242          BEQL    21$
                                50          08  A2  D0 00244          MOVL    8(CRT), R0                              ; 7149
                             0C A0          04  A3  91 00248          CMPB    4(PLC), 12(R0)
                                                27  13 0024D          BEQL    22$
                                52              62  D0 0024F          MOVL    (CRT), CRT                              ; 7150
                                                E4  11 00252          BRB     20$                                     ; 7147
                                                14  DD 00254 21$:     PUSHL   #20                                     ; 7156
                00000000G       00              01  FB 00256          CALLS   #1, GET_VM
                                52              50  D0 0025D          MOVL    R0, CRT
                00000000'       FF              62  0E 00260          INSQUE  (CRT), @QUEUE_HEADERS+12                ; 7157
                                50          08  A2  9E 00267          MOVAB   8(CRT), R0                              ; 7158
                             0C A2              50  D0 0026B          MOVL    R0, 12(CRT)
                             08 A2              50  D0 0026F          MOVL    R0, 8(CRT)
                                            10  A2  D4 00273          CLRL    16(CRT)                                 ; 7159
                                                0E  DD 00276 22$:     PUSHL   #14                                     ; 7161
                00000000G       00              01  FB 00278          CALLS   #1, GET_VM
                                55              50  D0 0027F          MOVL    R0, FID
                             0C B2              65  0E 00282          INSQUE  (FID), @12(CRT)                         ; 7162
                             08 A5              63  D0 00286          MOVL    (PLC), 8(FID)                           ; 7163
                             0C A5          04  A3  B0 0028A          MOVW    4(PLC), 12(FID)                         ; 7165
                                                0B  11 0028F          BRB     25$                                     ; 7166
                             10 A2              83  C0 00291 23$:     ADDL2   (PLC)+, 16(CRT)                         ; 7171
                                                12  11 00295          BRB     27$                                     ; 7134
                             10 A2          02  A3  C0 00297 24$:     ADDL2   2(PLC), 16(CRT)                         ; 7177
                                53              06  C0 0029C 25$:     ADDL2   #6, PLC                                 ; 7178
                                                08  11 0029F          BRB     27$                                     ; 7134
                             10 A2          02  A3  C0 002A1 26$:     ADDL2   2(PLC), 16(CRT)                         ; 7183
                                53              0A  C0 002A6          ADDL2   #10, PLC                                ; 7184
                                              FF56  31 002A9 27$:     BRW     15$                                     ; 7127
                                                14  DD 002AC 28$:     PUSHL   #20                                     ; 7203
                00000000G       00              01  FB 002AE          CALLS   #1, GET_VM
                                52              50  D0 002B5          MOVL    R0, CRT
                00000000'       FF              62  0E 002B8          INSQUE  (CRT), @QUEUE_HEADERS+8                 ; 7204
```

```
                                50        08    A2  9E 002BF          MOVAB   8(CRT), R0                          ; 7205
                            0C  A2              50  D0 002C3          MOVL    R0, 12(CRT)                         ;
                            08  A2              50  D0 002C7          MOVL    R0, 8(CRT)                          ;
                            10  A2        18    A6  D0 002CB          MOVL    24(FIB), 16(CRT)                    ; 7206
                                              0E  DD 002D0          PUSHL   #14                                 ; 7207
                    00000000G   00            01  FB 002D2          CALLS   #1, GET_VM                          ;
                            08  B2              60  0E 002D9          INSQUE  (FID), @8(CRT)                      ; 7208
                            08  A0        04    A6  D0 002DD          MOVL    4(FIB), 8(FID)                      ; 7209
                            0C  A0        08    A6  B0 002E2          MOVW    8(FIB), 12(FID)                     ; 7211
                                58 00000000'  EF  D0 002E7 29$:       MOVL    QUEUE_HEADERS+8, CRT                ; 7217
                            04  AE              57  D0 002EE          MOVL    HEADER, CUR_HDR                     ; 7218
                                53        04    A6  9E 002F2          MOVAB   4(R6), CUR_FID                      ; 7219
                                          18    A6  D5 002F6          TSTL    24(FIB)                             ; 7220
                                                03  12 002F9          BNEQ    30$                                 ;
                                             01C6  31 002FB          BRW     53$                                 ;
                            18  A6                54  D1 002FE 30$:   CMPL    TCOUNT, 24(FIB)                     ; 7223
                                                03  1F 00302          BLSSU   31$                                 ;
                                             01C8  31 00304          BRW     54$                                 ;
                                52        08    B8  0F 00307 31$:     REMQUE  @8(CRT), FID                        ; 7230
                    00000000'   EF              62  0E 0030B          INSQUE  (FID), QUEUE_HEADERS+16             ; 7231
                    00000000'   EF              58  D1 00312          CMPL    CRT, QUEUE_HEADERS+8                ; 7236
                                                25  13 00319          BEQL    33$                                 ;
                                          08    A2  9F 0031B          PUSHAB  8(FID)                              ; 7239
                                          3C    AE  9F 0031E          PUSHAB  EXT_HDR                             ;
                                                53  DD 00321          PUSHL   CUR_FID                             ;
                                          10    AE  DD 00323          PUSHL   CUR_HDR                             ;
                                FB99      CF        04  FB 00326          CALLS   #4, CREATE_EXTHDR                   ;
                                          6E        50  D0 0032B          MOVL    R0, STATUS                          ;
                                          03        6E  E8 0032E          BLBS    STATUS, 32$                         ; 7240
                                                 021E  31 00331          BRW     61$                                 ;
                                04  AE        38    AE  9E 00334 32$:     MOVAB   EXT_HDR, CUR_HDR                    ; 7241
                                          53        08  A2  9E 00339      MOVAB   8(FID), CUR_FID                     ; 7242
                                                    0C  AE  D6 0033D      INCL    HCOUNT                              ; 7243
                                50        18    A6        54  C3 00340 33$:   SUBL3   TCOUNT, 24(FIB), R0          ; 7247
                                          10    A8        50  D1 00345      CMPL    R0, 16(CRT)                         ;
                                                          04  1B 00349      BLEQU   34$                             ;
                                          50        10    A8  D0 0034B      MOVL    16(CRT), R0                         ;
                                04  AE                    50  D0 0034F 34$:   MOVL    R0, RCOUNT                  ;
                                          50 00000000'    EF  D0 00353      MOVL    CURRENT_MTL, R0                     ; 7258
                                14 00000000'    EF              06  E1 0035A      BBC     #6, QUAL+15, 35$            ; 7249
                                          0D 00000000'    EF    E9 00362      BLBC    QUAL+14, 35$                    ; 7250
                                0C  A2 00000000'    EF          91 00369      CMPB    QUAL+79, 12(FID)                ; 7251
                                                          03  13 00371      BEQL    35$                             ;
                                                       0139  31 00373      BRW     51$                             ;
                                          51        0C  A2  9A 00376 35$:   MOVZBL  12(FID), R1                     ; 7258
                                          55        30  A0  9A 0037A      MOVZBL  48(R0), R5                      ;
                                          51            55  C2 0037E      SUBL2   R5, R1                          ;
                            51      1F  A0              08  ED 00381      CMPZV   #0, #8, 31(R0), R1              ;
                                                          06  1A 00387      BGTRU   36$                             ;
                                          7E        7C  8F  9A 00389      MOVZBL  #124, -(SP)                     ; 7260
                                                          6F  11 0038D      BRB     39$                             ;
                                          51 00000000'    EF  D0 0038F 36$:   MOVL    CURRENT_MTL, R1             ; 7262
                                          50        0C  A2  9A 00396      MOVZBL  12(FID), R0                     ;
                                          55        30  A1  9A 0039A      MOVZBL  48(R1), R5                      ;
                                          50            55  C2 0039E      SUBL2   R5, R0                          ;
                                          59        34 A140  D0 003A1      MOVL    52(R1)[R0], VCB                 ;
                                    00000000'     EF        59  D0 003A6      MOVL    VCB, CURRENT_VCB            ;
```

```
                12      07  A9              02  E0 003AD          BBS     #2, 7(VCB), 37$                 ; 7264
                                        20  A9  9F 003B2          PUSHAB  32(VCB)                         ; 7266
                                            01  DD 003B5          PUSHL   #1                              ;
                                00000000G   8F  DD 003B7          PUSHL   #BACKUP$_NOVOLDATA              ;
                    00000000G   00          03  FB 003BD          CALLS   #3, LIB$SIGNAL                  ;
                                            14  DD 003C4 37$:     PUSHL   #20                             ; 7277
                    00000000G   00          01  FB 003C6          CALLS   #1, GET_VM                      ;
                                55          50  D0 003CD          MOVL    R0, EXT                         ;
                    00000000'   FF          65  0E 003D0          INSQUE  (EXT), @QUEUE_HEADERS           ; 7278
                          08    A5          59  D0 003D7          MOVL    VCB, 8(EXT)                     ; 7279
                                        10  A5  9F 003DB          PUSHAB  16(EXT)                         ; 7284
                                        0C  A5  9F 003DE          PUSHAB  12(EXT)                         ;
                                        10  AE  DD 003E1          PUSHL   RCOUNT                          ;
                              D364      CF      03  FB 003E4      CALLS   #3, STA_ALLOC_BEST              ;
                                        0C  A5  D5 003E9          TSTL    12(EXT)                         ; 7292
                                            0B  13 003EC          BEQL    38$                             ;
                                08      AE  0C  A5  D1 003EE      CMPL    12(EXT), RCOUNT                 ; 7293
                                            0B  1E 003F3          BGEQU   40$                             ;
                                    07      16  A6  E9 003F5      BLBC    22(FIB), 40$                    ;
                                    7E    0850  8F  3C 003F9 38$:   MOVZWL  #2128, -(SP)                   ; 7295
                                            54  11 003FE 39$:     BRB     47$                             ;
                                    5B      10  A5  D0 00400 40$:   MOVL    16(EXT), L                      ; 7304
                          50        10  A5  0C  A5  C1 00404 41$:   ADDL3   12(EXT), 16(EXT), R0           ; 7305
                                    50          5B  D1 0040A      CMPL    L, R0                           ;
                                                76  1E 0040D      BGEQU   50$                             ;
                          5A            50      5B  C3 0040F      SUBL3   L, R0, C                        ; 7307
                          1B        07  A9          01  E0 00413  BBS     #1, 7(VCB), 43$                 ; 7308
                             00000100   8F          5A  D1 00418  CMPL    C, #256                         ; 7311
                                                05  1B 0041F      BLEQU   42$                             ;
                                    5A    0100  8F  3C 00421      MOVZWL  #256, C                         ;
                                    7E          5A  7D 00426 42$:   MOVQ    C, -(SP)                    ; 7312
                                          0C    AE  DD 00429      PUSHL   CUR_HDR                         ;
                              D440      CF          03  FB 0042C  CALLS   #3, MAKE_POINTER1               ;
                                                0B  11 00431      BRB     44$                             ;
                                    7E          5A  7D 00433 43$:   MOVQ    C, -(SP)                    ; 7315
                                          0C    AE  DD 00436      PUSHL   CUR_HDR                         ;
                              D4A4      CF          03  FB 00439  CALLS   #3, MAKE_POINTER                ;
                                    6E          50  D0 0043E 44$:   MOVL    R0, STATUS                  ;
                                    05          6E  E9 00441      BLBC    STATUS, 46$                     ; 7317
                                    5B          5A  C0 00444      ADDL2   C, L                            ; 7319
                                                BB  11 00447 45$:   BRB     41$                             ;
                                    52    08    B8  0F 00449 46$:   REMQUE  @8(CRT), FID                    ; 7320
                                                08  1C 0044D      BVC     48$                             ;
                                    7E  08C8    8F  3C 0044F      MOVZWL  #2248, -(SP)                    ; 7322
                                              0175  31 00454 47$:   BRW     65$                             ;
                    00000000'       EF          62  0E 00457 48$:   INSQUE  (FID), QUEUE_HEADERS+16         ; 7325
                                    08    A2  9F 0045E      PUSHAB  8(FID)                          ; 7327
                                    3C    AE  9F 00461      PUSHAB  EXT_HDR                         ; 7326
                                          53  DD 00464      PUSHL   CUR_FID                         ; 7327
                                          10  AE  DD 00466      PUSHL   CUR_HDR                         ;
                              FA56      CF          04  FB 00469  CALLS   #4, CREATE_EXTHDR               ;
                                    6E          50  D0 0046E      MOVL    R0, STATUS                      ;
                                    03          6E  E8 00471      BLBS    STATUS, 49$                     ; 7328
                                              00DB  31 00474      BRW     61$                             ;
                                04  AE    38    AE  9E 00477 49$:   MOVAB   EXT_HDR, CUR_HDR            ; 7330
                                    53    08    A2  9E 0047C      MOVAB   8(FID), CUR_FID                 ; 7331
                                          0C    AE  D6 00480      INCL    HCOUNT                          ; 7332
```

```
                                    C2  11 00483          BRB     45$                                     ; 7305
                              7E    0C  A5  7D 00485 50$: MOVQ    12(EXT), -(SP)                          ; 7342
                              7E    0C  A2  9A 00489      MOVZBL  12(FID), -(SP)                          ;
                              50 00000000'  EF  D0 0048D  MOVL    CURRENT_MTL, R0                         ; 7341
                                    08  A0  DD 00494      PUSHL   8(R0)                                   ;
                        D7BD  CF        04  FB 00497      CALLS   #4, ADD_WINDOW_MAP                      ;
                              54    0C  A5  C0 0049C      ADDL2   12(EXT), TCOUNT                         ; 7348
                           08 AE    0C  A5  D1 004A0      CMPL    12(EXT), RCOUNT                         ; 7349
                                    17  1E 004A5          BGEQU   52$                                     ;
                           08 AE    0C  A5  C2 004A7      SUBL2   12(EXT), RCOUNT                         ; 7350
                                FF15    31 004AC          BRW     37$                                     ; 7269
                                    10  A8  DD 004AF 51$: PUSHL   16(CRT)                                ; 7355
                                    08  A0  DD 004B2      PUSHL   8(R0)                                   ;
                        D750  CF        02  FB 004B5      CALLS   #2, ADD_BLACKHOLE_MAP                   ;
                              54    10  A8  C0 004BA      ADDL2   16(CRT), TCOUNT                         ; 7356
                              58        68  D0 004BE 52$: MOVL    (CRT), CRT                              ; 7358
                                FE3A    31 004C1          BRW     30$                                     ; 7223
                              50    08  B8  0F 004C4 53$: REMQUE  @8(CRT), FID                           ; 7372
                  00000000'   EF        60  0E 004C8      INSQUE  (FID), QUEUE_HEADERS+16                 ; 7373
              07 00000000'    EF        06  E1 004CF 54$: BBC     #6, QUAL+15, 55$                        ; 7380
                              14 00000000'  EF  E8 004D7  BLBS    QUAL+14, 57$                            ; 7381
                        50    54        10  9C 004DE 55$: ROTL    #16, TCOUNT, R0                         ; 7384
                              01    07  A7  91 004E2      CMPB    7(HEADER), #1                           ; 7383
                                        06  12 004E6      BNEQ    56$                                     ;
                           12 A7            50  D0 004E8      MOVL    R0, 18(HEADER)                      ; 7384
                                            04  11 004EC      BRB     57$                                 ;
                           18 A7            50  D0 004EE 56$: MOVL    R0, 24(HEADER)                      ; 7385
                              50        38  AE  9E 004F2 57$: MOVAB   EXT_HDR, R0                        ; 7391
                              50        04  AE  D1 004F6      CMPL    CUR_HDR, R0                         ;
                                        21  12 004FA          BNEQ    59$                                 ;
                              0A 00000000'  EF  E9 004FC      BLBC    QUAL+14, 58$                        ; 7392
                        04    A3 00000000'  EF  91 00503      CMPB    QUAL+79, 4(CUR_FID)                 ;
                                        10  12 0050B          BNEQ    59$                                 ;
                                    04  AE  DD 0050D 58$: PUSHL   CUR_HDR                                ; 7395
                                        53  DD 00510          PUSHL   CUR_FID                             ;
                        D01B  CF        02  FB 00512          CALLS   #2, WRITE_HEADER                    ;
                              6E        50  D0 00517          MOVL    R0, STATUS                          ;
                              35        6E  E9 0051A          BLBC    STATUS, 61$                         ; 7396
              18 00000000'    EF        06  E1 0051D 59$:     BBC     #6, QUAL+15, 60$                    ; 7403
                              11 00000000'  EF  E9 00525      BLBC    QUAL+14, 60$                        ; 7404
                              50 00000000'  EF  D0 0052C      MOVL    CURRENT_MTL, R0                     ; 7405
                           1C A0 00000000'  EF  91 00533      CMPB    QUAL+79, 28(R0)                     ;
                                        19  12 0053B          BNEQ    62$                                 ;
                                        57  DD 0053D 60$:     PUSHL   HEADER                              ; 7408
              7E 00000000'  EF          18  C1 0053F          ADDL3   #24, CURRENT_MTL, -(SP)             ;
                    CFE6    CF        02  FB 00547          CALLS   #2, WRITE_HEADER                    ;
                              6E        50  D0 0054C          MOVL    R0, STATUS                          ;
                              04        6E  E8 0054F          BLBS    STATUS, 62$                         ; 7409
                                        6E  DD 00552 61$:     PUSHL   STATUS                              ;
                                        76  11 00554          BRB     65$                                 ;
                              02    07  A7  91 00556 62$:     CMPB    7(HEADER), #2                       ; 7415
                                        6E  12 0055A          BNEQ    64$                                 ;
              66 00000000'    EF        06  E1 0055C          BBC     #6, QUAL+15, 64$                    ;
                              5F 00000000'  EF  E8 00564      BLBS    QUAL+14, 64$                        ;
                              50    14  A7  9E 0056B          MOVAB   20(HEADER), R0                      ; 7422
                              01        60  91 0056F          CMPB    (R0), #1                            ;
                                        4A  12 00572          BNEQ    63$                                 ;
```

```
                                        01    A0  95 00574          TSTB    1(R0)                                          ; 7423
                                              45  12 00577          BNEQ    63$
                                  20    02    A0  B1 00579          CMPW    2(R0), #32                                     ; 7424
                                              3F  12 0057D          BNEQ    63$
                                        34    A7  95 0057F          TSTB    52(HEADER)                                     ; 7425
                                              3A  18 00582          BGEQ    63$
                                  04    42    A7  B1 00584          CMPW    66(HEADER), #4                                 ; 7426
                                              34  12 00588          BNEQ    63$
                                        47    A7  95 0058A          TSTB    71(HEADER)                                     ; 7427
                                              2F  12 0058D          BNEQ    63$
                                  04    44    A7  B1 0058F          CMPW    68(HEADER), #4                                 ; 7428
                                              29  12 00593          BNEQ    63$
                                  01    08    A6  91 00595          CMPB    8(FIB), #1                                     ; 7429
                                              23  12 00599          BNEQ    63$
                                  0B    20    BC  B1 0059B          CMPW    @P2, #11                                       ; 7430
                                              1D  12 0059F          BNEQ    63$
                                  50    20    AC  D0 005A1          MOVL    P2, R0                                         ; 7432
                FA49    CF        04    B0    0B  29 005A5          CMPC3   #11, @4(R0), P.AAU                             ; 7431
                                              10  12 005AC          BNEQ    63$
                      00000000'   EF    04    A6  D0 005AE          MOVL    4(FIB), DQF_QUOTA_FID                          ; 7436
                      00000000'   EF    08    A6  B0 005B6          MOVW    8(FIB), DQF_QUOTA_FID+4                        ; 7438
                                        0C BE44   9F 005BE 63$:     PUSHAB  @#COUNT[TCOUNT]                                ; 7444
                                        3C    A7  DD 005C2          PUSHL   60(HEADER)
                          CC75  CF              02  FB 005C5          CALLS   #2, DQF_MODIFY_USAGE
                                              01  DD 005CA 64$:     PUSHL   #1                                             ; 7450
                          F846  CF              01  FB 005CC 65$:     CALLS   #1, CREATE_CLEANUP                             ; 7451
                                                  04 005D1          RET

; Routine Size:  1490 bytes,    Routine Base:  CODE + 2E0E
```

```
;  5940          7452  1 %SBTTL 'STA_DEACCESS - deaccess QIO service routine'
;  5941          7453  1 ROUTINE STA_DEACCESS (EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)=
;  5942          7454  1
;  5943          7455  1 !++
;  5944          7456  1 !
;  5945          7457  1 ! FUNCTIONAL DESCRIPTION:
;  5946          7458  1 !       This routine executes IO$_DEACCESS in the standalone environment.
;  5947          7459  1 !
;  5948          7460  1 ! INPUT PARAMETERS:
;  5949          7461  1 !       As for $QIO(W) system service.
;  5950          7462  1 !
;  5951          7463  1 ! IMPLICIT INPUTS:
;  5952          7464  1 !       CURRENT_MTL     - Pointer to MTL for selected volume set.
;  5953          7465  1 !
;  5954          7466  1 ! OUTPUT PARAMETERS:
;  5955          7467  1 !       NONE
;  5956          7468  1 !
;  5957          7469  1 ! IMPLICIT OUTPUTS:
;  5958          7470  1 !       NONE
;  5959          7471  1 !
;  5960          7472  1 ! ROUTINE VALUE:
;  5961          7473  1 !       Completion status.
;  5962          7474  1 !
;  5963          7475  1 ! SIDE EFFECTS:
;  5964          7476  1 !       NONE
;  5965          7477  1 !
;  5966          7478  1 !--
;  5967          7479  1
;  5968          7480  2 BEGIN
;  5969          7481  2
;  5970          7482  2 MAP
;  5971          7483  2         P1:             REF BBLOCK;     ! Descriptor for FIB
;  5972          7484  2
;  5973          7485  2 LABEL
;  5974          7486  2         WRITE_ACL;                      ! ACL writing loop
;  5975          7487  2
;  5976          7488  2 LOCAL
;  5977          7489  2         WRITE_FLAG,                     ! Flag indicating header must be written
;  5978          7490  2         ACL_POINTER     : REF BBLOCK,   ! Address of current ACL segment
;  5979          7491  2         ACE_POINTER     : REF BBLOCK,   ! Current ACE in core
;  5980          7492  2         ACE             : REF BBLOCK,   ! Current ACE in header
;  5981          7493  2         ACL_LENGTH,                     ! Size of current header segment ACL
;  5982          7494  2         LOCAL_HEADER    : BBLOCK [512], ! Local header storage
;  5983          7495  2         HEADER          : REF BBLOCK,   ! Pointer to current header segment
;  5984          7496  2         HEADER_FID      : BBLOCK [6],   ! File-id of current header
;  5985          7497  2         EXT_HEADER_FID  : BBLOCK [6],   ! Extension header file-ID
;  5986          7498  2         CRT             : REF BBLOCK,   ! Pointer to create list entry
;  5987          7499  2         FID             : REF BBLOCK,   ! Pointer to FID entry
;  5988          7500  2         RVN,                            ! Current RVN we are working on
;  5989          7501  2         STATUS;                         ! Status return
;  5990          7502  2
;  5991          7503  2
;  5992          7504  2 ! Check that a file is accessed.
;  5993          7505  2 !
;  5994          7506  2 IF .CURRENT_MTL[MTL_WINDOW] EQL 0 THEN RETURN SS$_FILNOTACC;
;  5995          7507  2
;  5996          7508  2 STATUS = SS$_NORMAL;                  ! Assume success
```

```
;  5997          7509  2
;  5998          7510  2 ! Write attributes if specified.
;  5999          7511  2 !
;  6000          7512  2 WRITE_FLAG = FALSE;
;  6001          7513  2 HEADER = .CURRENT_MTL[MTL_HEADER];
;  6002          7514  2 IF .P5 NEQ 0
;  6003          7515  2 THEN
;  6004          7516  3     BEGIN
;  6005          7517  3     STATUS = WRITE_ATTRIBUTES (.HEADER, .P5, (IF .P1 EQL 0 THEN 0 ELSE .P1[DSC$A_POINTER]));
;  6006          7518  3     IF .STATUS
;  6007          7519  3     THEN WRITE_FLAG = TRUE;
;  6008          7520  2     END;
;  6009          7521  2
;  6010          7522  2 ! Write the highwater mark if it has been modified. This is not done for
;  6011          7523  2 ! multi-volume files in /VOLUME restores, nor for multi-volume save sets.
;  6012          7524  2 !
;  6013          7525  2
;  6014          7526  2 IF .BBLOCK [.CURRENT_MTL[MTL_WINDOW], WCB_CUR_HWM]
;  6015          7527  2     NEQ .BBLOCK [.CURRENT_MT[[MTL_WINDOW], WCB_SET_HWM]
;  6016          7528  2 AND
;  6017          7529  3     BEGIN
;  6018          7530  4     IF (.QUAL[QUAL_OF11] AND .QUAL[QUAL_VOLU])
;  6019          7531  3     OR .QUAL[QUAL_OSAV]
;  6020          7532  3     THEN .HEADER[FH2$W_SEG_NUM] EQL 0
;  6021          7533  3         AND (.HEADER[FH2$W_EX_FIDNUM] OR .HEADER[FH2$W_EX_FIDRVN]) EQL 0
;  6022          7534  3     ELSE TRUE
;  6023          7535  3     END
;  6024          7536  2 AND .HEADER[FH2$B_STRUCLEV] EQL 2
;  6025          7537  2 AND .HEADER[FH2$B_IDOFFSET] GEQU ($BYTEOFFSET (FH2$L_HIGHWATER) + 4) / 2
;  6026          7538  2 THEN
;  6027          7539  3     BEGIN
;  6028          7540  3     HEADER[FH2$L_HIGHWATER] = .BBLOCK [.CURRENT_MTL[MTL_WINDOW], WCB_CUR_HWM];
;  6029          7541  3     WRITE_FLAG = TRUE;
;  6030          7542  2     END;
;  6031          7543  2
;  6032          7544  2 ! Write the header if necessary.
;  6033          7545  2 !
;  6034          7546  2 IF .WRITE_FLAG
;  6035          7547  2 THEN STATUS = WRITE_HEADER (CURRENT_MTL[MTL_FID], .HEADER);
;  6036          7548  2
;  6037          7549  2
;  6038          7550  2 ! Write out the file's ACL if necessary.
;  6039          7551  2 !
;  6040          7552  2 IF .CURRENT_MTL[MTL_ACLFL] NEQA CURRENT_MTL[MTL_ACLFL]
;  6041          7553  2 AND .CURRENT_MTL[MT[_NEW_ACL]
;  6042          7554  2 AND .STATUS
;  6043          7555  2 THEN
;  6044          7556  3     WRITE_ACL:  BEGIN
;  6045          7557  3     ACL_POINTER = .CURRENT_MTL[MTL_ACLFL];
;  6046          7558  3     ACE_POINTER = ACL_POINTER[ACL$[_LIST];
;  6047          7559  3     RVN = .CURRENT_MT[[MTL_FID_RVN];
;  6048          7560  3     WHILE 1
;  6049          7561  3     DO
;  6050          7562  4         BEGIN
;  6051          7563  5         ACL_LENGTH = (.HEADER[FH2$B_RSOFFSET] -
;  6052          7564  5                       .HEADER[FH2$B_MPOFFSET] -
;  6053          7565  5                       MAXU (.HEADER[FH2$B_MAP_INUSE],
```

```
;6054    7566  6                                 (IF .HEADER[FH2$W_SEG_NUM] EQL 0
;6055    7567  6                                  THEN FM2$C_LENGTH3/2
;6056    7568  4                                  ELSE 0))) * 2;
;6057    7569  4          IF .ACE_POINTER[ACE$B_SIZE] LEQ .ACL_LENGTH THEN EXITLOOP;
;6058    7570  4          CH$MOVE (6, HEADER[FH2$W_EXT_FID], EXT_HEADER_FID);
;6059    7571  4          IF .EXT_HEADER_FID[FID$B_RVN] EQL 0
;6060    7572  4          THEN EXT_HEADER_FID[FID$B_RVN] = .RVN
;6061    7573  4          ELSE RVN = .EXT_HEADER_FID[FID$B_RVN];
;6062    7574  4          STATUS = READ_HEADER (EXT_HEADER_FID, LOCAL_HEADER);
;6063    7575  4          IF NOT .STATUS
;6064    7576  4          THEN
;6065    7577  5              BEGIN
;6066    7578  5              ACL_DELETEACL ();
;6067    7579  5              LEAVE WRITE_ACL;
;6068    7580  4              END;
;6069    7581  4          HEADER = LOCAL_HEADER;
;6070    7582  3          END;
;6071    7583  3      CH$MOVE (6, HEADER[FH2$W_FID], HEADER_FID);
;6072    7584  3      HEADER_FID[FID$B_RVN] = .RVN;
;6073    7585  3      HEADER[FH2$B_ACOFFSET] = .HEADER[FH2$B_RSOFFSET] - .ACL_LENGTH / 2;
;6074    7586  3      ACE = .HEADER + .HEADER[FH2$B_ACOFFSET] * 2;
;6075    7587  3      WHILE 1
;6076    7588  3      DO
;6077    7589  4          BEGIN
;6078    7590  4          CH$MOVE (.ACE_POINTER[ACE$B_SIZE], .ACE_POINTER, .ACE);
;6079    7591  4          ACE = .ACE + .ACE_POINTER[ACE$B_SIZE];
;6080    7592  4          ACL_LENGTH = .ACL_LENGTH - .ACE_POINTER[ACE$B_SIZE];
;6081    7593  4          ACE_POINTER = .ACE_POINTER + .ACE_POINTER[ACE$B_SIZE];
;6082    7594  4          IF .ACE_POINTER GEQA .ACL_POINTER + .ACL_POINTER[ACL$W_SIZE]
;6083    7595  4          THEN
;6084    7596  5              BEGIN
;6085    7597  5              ACL_POINTER = .ACL_POINTER[ACL$L_FLINK];
;6086    7598  5              IF .ACL_POINTER EQLA CURRENT_MTL[MTL_ACLFL] THEN EXITLOOP;
;6087    7599  5              ACE_POINTER = ACL_POINTER[ACL$L_LIST];
;6088    7600  4              END;
;6089    7601  4          IF .ACE_POINTER[ACE$B_SIZE] GTR .ACL_LENGTH
;6090    7602  4          THEN
;6091    7603  5              BEGIN
;6092    7604  5              CRT = .QUEUE_HEADERS[2];
;6093    7605  5              IF REMQUE (.CRT[CRT_FID_FQHDR], FID)                ! Get FID entry
;6094    7606  5              THEN
;6095    7607  6                  BEGIN
;6096    7608  6                  ACL_DELETEACL ();
;6097    7609  6                  STATUS = SS$_HEADERFULL;
;6098    7610  6                  LEAVE WRITE_ACL;
;6099    7611  5                  END;
;6100    7612  5              INSQUE (.FID, QUEUE_HEADERS[4]);
;6101    7613  5              STATUS = CREATE_EXTHDR (.HEADER, HEADER_FID, LOCAL_HEADER, FID[CRT_FID]);
;6102    7614  5              IF NOT .STATUS
;6103    7615  5              THEN
;6104    7616  6                  BEGIN
;6105    7617  6                  ACL_DELETEACL ();
;6106    7618  6                  LEAVE WRITE_ACL;
;6107    7619  5                  END;
;6108    7620  5              IF .HEADER NEQA LOCAL_HEADER
;6109    7621  5              THEN
;6110    7622  6                  BEGIN
```

```
; 6111          7623  6                 STATUS = WRITE_HEADER (HEADER_FID, .HEADER);
; 6112          7624  6                 IF NOT .STATUS
; 6113          7625  6                 THEN
; 6114          7626  7                     BEGIN
; 6115          7627  7                     ACL_DELETEACL ();
; 6116          7628  7                     LEAVE WRITE_ACL;
; 6117          7629  6                     END;
; 6118          7630  5                 END;
; 6119          7631  5             HEADER = LOCAL_HEADER;
; 6120          7632  5             CH$MOVE (6, HEADER[FH2$W_FID], HEADER_FID);
; 6121          7633  5             HEADER_FID[FID$B_RVN] = .RVN;
; 6122          7634  6             ACL_LENGTH = (.HEADER[FH2$B_RSOFFSET] -
; 6123          7635  6                           .HEADER[FH2$B_MPOFFSET] -
; 6124          7636  5                           .HEADER[FH2$B_MAP_INUSE]) * 2;
; 6125          7637  5             HEADER[FH2$B_ACOFFSET] = .HEADER[FH2$B_RSOFFSET] - .ACL_LENGTH / 2;
; 6126          7638  5             ACE = .HEADER + .HEADER[FH2$B_ACOFFSET] * 2;
; 6127          7639  4             END;
; 6128          7640  3         END;
; 6129          7641  3
; 6130          7642  3 ! Recover any unused ACL space from this header by sliding the ACL down
; 6131          7643  3 ! to the end of the header. Clear the odd byte at the end of the ACL
; 6132          7644  3 ! if there is one.
; 6133          7645  3
; 6134          7646  3     IF TESTBITSC (ACL_LENGTH<0,1>) THEN (.ACE)<0,8> = 0;
; 6135          7647  3     CH$MOVE ((.HEADER[FH2$B_RSOFFSET] - .HEADER[FH2$B_ACOFFSET]) * 2 - .ACL_LENGTH,
; 6136          7648  3              .HEADER + .HEADER[FH2$B_ACOFFSET] * 2,
; 6137          7649  3              .HEADER + .HEADER[FH2$B_ACOFFSET] * 2 + .ACL_LENGTH);
; 6138          7650  3     CH$FILL (0, .ACL_LENGTH, .HEADER + .HEADER[FH2$B_ACOFFSET] * 2);
; 6139          7651  3     HEADER[FH2$B_ACOFFSET] = .HEADER[FH2$B_ACOFFSET] + .ACL_LENGTH / 2;
; 6140          7652  3     STATUS = WRITE_HEADER (HEADER_FID, .HEADER);
; 6141          7653  3     IF NOT .STATUS
; 6142          7654  3     THEN
; 6143          7655  4         BEGIN
; 6144          7656  4         ACL_DELETEACL ();
; 6145          7657  4         LEAVE WRITE_ACL;
; 6146          7658  3         END;
; 6147          7659  2     END;                                    ! End of WRITE_ACL loop
; 6148          7660  2
; 6149          7661  2
; 6150          7662  2 ! Do the final cleanup for the deaccess operation.
; 6151          7663  2 !
; 6152          7664  2
; 6153          7665  2
; 6154          7666  2 ! Free the create list.
; 6155          7667  2 !
; 6156          7668  2 UNTIL REMQUE (.QUEUE_HEADERS[2], CRT) DO
; 6157          7669  3     BEGIN
; 6158          7670  3     LOCAL
; 6159          7671  3         Q:              REF BBLOCK;     ! Pointer to FID block
; 6160          7672  3
; 6161          7673  3
; 6162          7674  3     ! Free the FID blocks.
; 6163          7675  3     !
; 6164          7676  3     UNTIL REMQUE (.CRT[CRT_FID_FQHDR], Q) DO FREE_VM (CRT_S_FID, .Q);
; 6165          7677  3
; 6166          7678  3
; 6167          7679  3     ! Deallocate the create list entry.
```

```
;  6168          7680  3     !
;  6169          7681  3     FREE_VM (CRT_S_BLOCKS, .CRT);
;  6170          7682  2     END;
;  6171          7683  2
;  6172          7684  2
;  6173          7685  2 ! Free the used file ID list.
;  6174          7686  2 !
;  6175          7687  2 UNTIL REMQUE (.QUEUE_HEADERS[4], CRT) DO
;  6176          7688  3     BEGIN
;  6177          7689  3
;  6178          7690  3
;  6179          7691  3     ! If the deaccess failed, write a deleted file header.
;  6180          7692  3     !
;  6181          7693  3     IF NOT .STATUS
;  6182          7694  3     THEN
;  6183          7695  4         BEGIN
;  6184          7696  4         CREATE_DELHDR (CRT[CRT_FID], LOCAL_HEADER);
;  6185          7697  4         WRITE_HEADER (CRT[CRT_FID], LOCAL_HEADER);
;  6186          7698  3         END;
;  6187          7699  2     END;
;  6188          7700  2
;  6189          7701  2
;  6190          7702  2 ! Release the window.
;  6191          7703  2 !
;  6192          7704  2 DELETE_WINDOW (.CURRENT_MTL[MTL_WINDOW]);
;  6193          7705  2 CURRENT_MTL[MTL_WINDOW] = 0;
;  6194          7706  2
;  6195          7707  2
;  6196          7708  2 ! Indicate that there is no ACL associated with the file.
;  6197          7709  2 !
;  6198          7710  2 CURRENT_MTL[MTL_ACLFL] = CURRENT_MTL[MTL_ACLBL] = 0;
;  6199          7711  2
;  6200          7712  2
;  6201          7713  2 ! Return with the final status.
;  6202          7714  2 !
;  6203          7715  2 RETURN .STATUS;
;  6204          7716  2
;  6205          7717  1 END;


                                OFFC 00000 STA_DEACCESS:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 7453
                     5E     FDE4   CE  9E 00002    MOVAB   -540(SP), SP                          ;
                     50 00000000'  EF  D0 00007    MOVL    CURRENT_MTL, R0                       ; 7506
                              08   A0  D5 0000E    TSTL    8(R0)                                 ;
                                   05  12 00011    BNEQ    1$                                    ;
                     50       AC   8F  9A 00013    MOVZBL  #172, R0                              ;
                                   04  00017    RET                                           ;
                     6E            01  D0 00018 1$:  MOVL    #1, STATUS                        ; 7508
                                   53  D4 0001B    CLRL    WRITE_FLAG                            ; 7512
                     56       0C   A0  D0 0001D    MOVL    12(R0), HEADER                        ; 7513
                              2C   AC  D5 00021    TSTL    P5                                    ; 7514
                                   20  13 00024    BEQL    4$                                    ;
                     50       1C   AC  D0 00026    MOVL    P1, R0                                ; 7517
```

```
                                        04   12 0002A          BNEQ    2$                                      ;
                                        7E   D4 0002C          CLRL    -(SP)                                   ;
                                        03   11 0002E          BRB     3$                                      ;
                                  04    A0   DD 00030 2$:      PUSHL   4(R0)                                   ;
                                  2C    AC   DD 00033 3$:      PUSHL   P5                                      ;
                                        56   DD 00036          PUSHL   HEADER                                  ;
                          0000V CF            03   FB 00038          CALLS   #3, WRITE_ATTRIBUTES              ;
                                6E            50   D0 0003D          MOVL    R0, STATUS                        ;
                                03            6E   E9 00040          BLBC    STATUS, 4$                        ;  7518
                                53            01   D0 00043          MOVL    #1, WRITE_FLAG                    ;  7519
                                51 00000000'  EF   D0 00046 4$:      MOVL    CURRENT_MTL, R1                   ;  7526
                                50      08    A1   D0 0004D          MOVL    8(R1), R0                         ;
                             10 A0      0C    A0   D1 00051          CMPL    12(R0), 16(R0)                    ;  7527
                                           3C   13 00056          BEQL    8$                                ;
                          07 00000000'  EF            06   E1 00058          BBC     #6, QUAL+15, 5$           ;  7530
                                08 00000000'  EF      E8 00060          BLBS    QUAL+14, 6$                       ;
                                   00000000'  EF      95 00067 5$:      TSTB    QUAL+15                           ;  7531
                                              12   18 0006D          BGEQ    7$                                ;
                                        04    A6   B5 0006F 6$:      TSTW    4(HEADER)                         ;  7532
                                              20   12 00072          BNEQ    8$                                ;
                                52      0E    A6   3C 00074          MOVZWL  14(HEADER), R2                    ;  7533
                                54      12    A6   3C 00078          MOVZWL  18(HEADER), R4                    ;
                                52            54   C8 0007C          BISL2   R4, R2                            ;
                                              13   12 0007F          BNEQ    8$                                ;
                                02      07    A6   91 00081 7$:      CMPB    7(HEADER), #2                     ;  7536
                                              0D   12 00085          BNEQ    8$                                ;
                                28            66   91 00087          CMPB    (HEADER), #40                     ;  7537
                                              08   1F 0008A          BLSSU   8$                                ;
                             4C A6      0C    A0   D0 0008C          MOVL    12(R0), 76(HEADER)                ;  7540
                                53            01   D0 00091          MOVL    #1, WRITE_FLAG                    ;  7541
                                0D            53   E9 00094 8$:      BLBC    WRITE_FLAG, 9$                    ;  7546
                                              56   DD 00097          PUSHL   HEADER                            ;  7547
                                        18    A1   9F 00099          PUSHAB  24(R1)                            ;
                          CEBF  CF            02   FB 0009C          CALLS   #2, WRITE_HEADER                  ;
                                6E            50   D0 000A1          MOVL    R0, STATUS                        ;
                                50 00000000'  EF   D0 000A4 9$:      MOVL    CURRENT_MTL, R0                   ;  7552
                                51      10    A0   9E 000AB          MOVAB   16(R0), R1                        ;
                                51      10    A0   D1 000AF          CMPL    16(R0), R1                        ;
                                              03   12 000B3          BNEQ    11$                               ;
                                           0199   31 000B5 10$:      BRW     29$                               ;
                       F8    31 A0            01   E1 000B8 11$:      BBC     #1, 49(R0), 10$                   ;  7553
                                F5            6E   E9 000BD          BLBC    STATUS, 10$                       ;  7554
                                59      10    A0   D0 000C0          MOVL    16(R0), ACL_POINTER               ;  7557
                                58      0C    A9   9E 000C4          MOVAB   12(R9), ACE_POINTER               ;  7558
                             04 AE      1C    A0   9A 000C8          MOVZBL  28(R0), RVN                       ;  7559
                                52      03    A6   9A 000CD 12$:     MOVZBL  3(HEADER), R2                     ;  7564
                                50      01    A6   9A 000D1          MOVZBL  1(HEADER), R0                     ;
                                52            50   C2 000D5          SUBL2   R0, R2                            ;
                                        04    A6   B5 000D8          TSTW    4(HEADER)                         ;  7566
                                              05   12 000DB          BNEQ    13$                               ;
                                51            04   D0 000DD          MOVL    #4, R1                            ;  7567
                                              02   11 000E0          BRB     14$                               ;
                                              51   D4 000E2 13$:     CLRL    R1                                ;  7566
                                50      3A    A6   9A 000E4 14$:     MOVZBL  58(HEADER), R0                    ;
                                51            50   D1 000E8          CMPL    R0, R1                            ;
                                              03   1E 000EB          BGEQU   15$                               ;
                                50            51   D0 000ED          MOVL    R1, R0                            ;
```

```
                                52              50  C2 000F0 15$:    SUBL2   R0, R2                                          ; 7565
                             57 52              01  78 000F3         ASHL    #1, R2, ACL_LENGTH                              ; 7568
                     57      68 08              00  ED 000F7         CMPZV   #0, #8, (ACE_POINTER), ACL_LENGTH               ; 7569
                                                31  15 000FC         BLEQ    19$                                             ;
                     0C  AE     0E  A6          06  28 000FE         MOVC3   #6, 14(HEADER), EXT_HEADER_FID                  ; 7570
                                        10  AE  95 00104             TSTB    EXT_HEADER_FID+4                                ; 7571
                                            07  12 00107             BNEQ    16$                                             ;
                                    10  AE  04  AE 90 00109         MOVB    RVN, EXT_HEADER_FID+4                           ; 7572
                                            05  11 0010E             BRB     17$                                             ;
                                    04  AE  10  AE 9A 00110 16$:     MOVZBL  EXT_HEADER_FID+4, RVN                           ; 7573
                                        1C  AE 9F 00115 17$:         PUSHAB  LOCAL_HEADER                                    ; 7574
                                        10  AE 9F 00118             PUSHAB  EXT_HEADER_FID                                  ;
                                CDB0    CF     02  FB 0011B         CALLS   #2, READ_HEADER                                 ;
                                        6E     50  D0 00120         MOVL    R0, STATUS                                      ;
                                        03     6E  E8 00123         BLBS    STATUS, 18$                                     ; 7575
                                               0121 31 00126        BRW     28$                                             ;
                                        56  1C  AE 9E 00129 18$:     MOVAB   LOCAL_HEADER, HEADER                            ; 7581
                                                9E 11 0012D         BRB     12$                                             ; 7560
                     14  AE     08  A6          06  28 0012F 19$:    MOVC3   #6, 8(HEADER), HEADER_FID                       ; 7583
                                18  AE  04  AE 90 00135             MOVB    RVN, HEADER_FID+4                               ; 7584
                             50     57     02  C7 0013A 20$:        DIVL3   #2, ACL_LENGTH, R0                              ; 7585
                     02  A6      03  A6     50  83 0013E            SUBB3   R0, 3(HEADER), 2(HEADER)                        ;
                                        50  02  A6 9A 00144         MOVZBL  2(HEADER), R0                                   ; 7586
                                08  AE     6640 3E 00148            MOVAW   (HEADER)[R0], ACE                               ;
                                        50      68  9A 0014D 21$:    MOVZBL  (ACE_POINTER), R0                               ; 7590
                     08  BE                68      50  28 00150     MOVC3   R0, (ACE_POINTER), @ACE                         ;
                                        50      68  9A 00155         MOVZBL  (ACE_POINTER), R0                               ; 7591
                                08  AE         50  C0 00158         ADDL2   R0, ACE                                         ;
                                        50      68  9A 0015C         MOVZBL  (ACE_POINTER), R0                               ; 7592
                                        57      50  C2 0015F         SUBL2   R0, ACL_LENGTH                                  ;
                                        50      68  9A 00162         MOVZBL  (ACE_POINTER), R0                               ; 7593
                                        58      50  C0 00165         ADDL2   R0, ACE_POINTER                                 ;
                                        50  08  A9  3C 00168         MOVZWL  8(ACL_POINTER), R0                              ; 7594
                                        50      59  C0 0016C         ADDL2   ACL_POINTER, R0                                 ;
                                        50      58  D1 0016F         CMPL    ACE_POINTER, R0                                 ;
                                                17  1F 00172         BLSSU   23$                                             ;
                                        59      69  D0 00174         MOVL    (ACL_POINTER), ACL_POINTER                      ; 7597
                         50 00000000'   EF      10  C1 00177         ADDL3   #16, CURRENT_MTL, R0                            ; 7598
                                        50      59  D1 0017F         CMPL    ACL_POINTER, R0                                 ;
                                                03  12 00182         BNEQ    22$                                             ;
                                               0085 31 00184        BRW     26$                                             ;
                                        58  0C  A9  9E 00187 22$:     MOVAB   12(R9), ACE_POINTER                             ; 7599
                 57      68             08          00  ED 0018B 23$:    CMPZV   #0, #8, (ACE_POINTER), ACL_LENGTH           ; 7601
                                                    BB  15 00190         BLEQ    21$                                         ;
                                        5A 00000000'    EF  D0 00192     MOVL    QUEUE_HEADERS+8, CRT                        ; 7604
                                        5B      08      BA  0F 00199     REMQUE  @8(CRT), FID                                ; 7605
                                                        0F  1C 0019D     BVC     24$                                         ;
                             00000000G  00              00  FB 0019F     CALLS   #0, ACL_DELETEACL                           ; 7608
                                        6E      08C8    8F  3C 001A6     MOVZWL  #2248, STATUS                               ; 7609
                                                0A3     31  001AB        BRW     29$                                         ; 7610
                             00000000'  EF              6B  0E 001AE 24$:    INSQUE  (FID), QUEUE_HEADERS+16                 ; 7612
                                                08      AB  9F 001B5     PUSHAB  8(FID)                                      ; 7613
                                                20      AE  9F 001B8     PUSHAB  LOCAL_HEADER                                ;
                                                1C      AE  9F 001BB     PUSHAB  HEADER_FID                                  ;
                                                        56  DD 001BE     PUSHL   HEADER                                      ;
                                 F72D   CF              04  FB 001C0     CALLS   #4, CREATE_EXTHDR                           ;
                                        6E              50  D0 001C5     MOVL    R0, STATUS                                  ;
```

```
                                                
```

```
                          08   AA  9F 00294          PUSHAB  8(CRT)                                          ;
                CCC4      CF        02 FB 00297          CALLS   #2, WRITE_HEADER                           ;
                                    DC 11 0029C          BRB     32$                                        ; 7687
                          50 00000000'  EF  D0 0029E 33$:  MOVL    CURRENT_MTL, R0                          ; 7704
                          08   A0  DD 002A5          PUSHL   8(R0)                                          ;
                D35F      CF        01 FB 002A8          CALLS   #1, DELETE_WINDOW                          ;
                          50 00000000'  EF  D0 002AD          MOVL    CURRENT_MTL, R0                      ; 7705
                          08   A0  D4 002B4          CLRL    8(R0)                                          ;
                          10   A0  7C 002B7          CLRQ    16(R0)                                         ; 7710
                          50        6E  D0 002BA          MOVL    STATUS, R0                                 ; 7715
                                    04 002BD          RET                                                    ; 7717

; Routine Size:  702 bytes,    Routine Base:  CODE + 33E0
```

```
;  6207          7718  1 %SBTTL 'STA_MODIFY - modify QIO service routine'
;  6208          7719  1 ROUTINE STA_MODIFY (EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)=
;  6209          7720  1
;  6210          7721  1 !++
;  6211          7722  1 !
;  6212          7723  1 ! FUNCTIONAL DESCRIPTION:
;  6213          7724  1 !       This routine executes IO$_MODIFY in the standalone environment.
;  6214          7725  1 !
;  6215          7726  1 ! INPUT PARAMETERS:
;  6216          7727  1 !       As for $QIO(W) system service.  However, a nonzero P6 points to
;  6217          7728  1 !       OUTPUT_ATTBUF, which indicates that the IO$_MODIFY refers to a
;  6218          7729  1 !       file on an image output volume.
;  6219          7730  1 !
;  6220          7731  1 ! IMPLICIT INPUTS:
;  6221          7732  1 !       CURRENT_MTL     - Pointer to MTL for selected volume set.
;  6222          7733  1 !
;  6223          7734  1 ! OUTPUT PARAMETERS:
;  6224          7735  1 !       NONE
;  6225          7736  1 !
;  6226          7737  1 ! IMPLICIT OUTPUTS:
;  6227          7738  1 !       NONE
;  6228          7739  1 !
;  6229          7740  1 ! ROUTINE VALUE:
;  6230          7741  1 !       Completion status.
;  6231          7742  1 !
;  6232          7743  1 ! SIDE EFFECTS:
;  6233          7744  1 !       NONE
;  6234          7745  1 !
;  6235          7746  1 !--
;  6236          7747  1
;  6237          7748  2 BEGIN
;  6238          7749  2 MAP
;  6239          7750  2         P1:             REF BBLOCK;     ! Descriptor for FIB
;  6240          7751  2 LOCAL
;  6241          7752  2         FIB:            REF BBLOCK,     ! Pointer to FIB
;  6242          7753  2         STATUS,                         ! Status variable
;  6243          7754  2         RVN,                            ! RVN of current header
;  6244          7755  2         HEADER:         REF BBLOCK,     ! Pointer to file header
;  6245          7756  2         FID:            REF BBLOCK,     ! Pointer to current file ID
;  6246          7757  2         EXT_FILE_ID:    BBLOCK [FID$C_LENGTH], ! FID of extension header
;  6247          7758  2         LOCAL_HEADER:   BBLOCK [512];   ! Local buffer for extension header
;  6248          7759  2
;  6249          7760  2 ! If the FIB descriptor is present, get the address of the FIB.
;  6250          7761  2 !
;  6251          7762  2 IF .P1 NEQ 0 THEN FIB = .P1[DSC$A_POINTER];
;  6252          7763  2
;  6253          7764  2 ! Get the file ID if the file is not open.
;  6254          7765  2 !
;  6255          7766  2 HEADER = .CURRENT_MTL[MTL_HEADER];
;  6256          7767  2 FID = CURRENT_MTL[MTL_FID];
;  6257          7768  2 IF .CURRENT_MTL[MTL_WINDOW] EQL 0
;  6258          7769  2 THEN
;  6259          7770  3     BEGIN
;  6260          7771  3     IF .P1 EQL 0 THEN RETURN SS$_BADPARAM;
;  6261          7772  3     FID[FID$W_NUM] = .FIB[FIB$W_FID_NUM];
;  6262          7773  3     FID[FID$W_SEQ] = .FIB[FIB$W_FID_SEQ];
;  6263          7774  3     FID[FID$W_RVN] = .FIB[FIB$W_FID_RVN];
```

```
;  6264          7775  3     STATUS = READ_HEADER(.FID, .HEADER);
;  6265          7776  3     IF NOT .STATUS THEN RETURN .STATUS;
;  6266          7777  3     CURRENT_MTL[MTL_NEW_ACL] = 0;
;  6267          7778  2     END;
;  6268          7779  2 RVN = .FID[FID$B_RVN];
;  6269          7780  2
;  6270          7781  2
;  6271          7782  2 ! Iterate over the headers of a multi-header file.
;  6272          7783  2 !
;  6273          7784  2
;  6274          7785  2 WHILE TRUE
;  6275          7786  2 DO
;  6276          7787  3     BEGIN
;  6277          7788  3
;  6278          7789  3 ! Write attributes.
;  6279          7790  3 !
;  6280          7791  3     STATUS = WRITE_ATTRIBUTES (.HEADER, .P5, (IF .P1 EQL 0 THEN 0 ELSE .P1[DSC$A_POINTER]));
;  6281          7792  3     IF NOT .STATUS THEN RETURN .STATUS;
;  6282          7793  3
;  6283          7794  3
;  6284          7795  3 ! Rewrite header.
;  6285          7796  3 !
;  6286          7797  3     STATUS = WRITE_HEADER(.FID, .HEADER);
;  6287          7798  3     IF NOT .STATUS THEN RETURN .STATUS;
;  6288          7799  3
;  6289          7800  3
;  6290          7801  3 ! Quota table maintenance.  The file header must be charged.
;  6291          7802  3 !
;  6292          7803  3     IF
;  6293          7804  3         .HEADER[FH2$B_STRUCLEV] EQL 2 AND NOT .QUAL[QUAL_VOLU] AND
;  6294          7805  3         .P6 NEQ 0 AND
;  6295          7806  4         (.FIB[FIB$W_FID_NUM] GEQU FID$C_MFD OR .FIB[FIB$B_FID_NMX] NEQ 0)
;  6296          7807  3     THEN
;  6297          7808  3         DQF_MODIFY_USAGE(.HEADER[FH2$L_FILEOWNER], 1);
;  6298          7809  3
;  6299          7810  3 ! Read and modify the next extension file header.
;  6300          7811  3 ! Get clean file number and RVN.
;  6301          7812  3 !
;  6302          7813  3     IF .HEADER[FH2$B_STRUCLEV] EQL 2
;  6303          7814  3     THEN
;  6304          7815  4         BEGIN
;  6305          7816  4         EXT_FILE_ID[FID$W_NUM] = .HEADER[FH2$W_EX_FIDNUM];
;  6306          7817  4         EXT_FILE_ID[FID$W_SEQ] = .HEADER[FH2$W_EX_FIDSEQ];
;  6307          7818  4         EXT_FILE_ID[FID$W_RVN] = .HEADER[FH2$W_EX_FIDRVN];
;  6308          7819  4         END
;  6309          7820  3     ELSE
;  6310          7821  4         BEGIN
;  6311          7822  4         LOCAL MAP_POINTER: REF BBLOCK;
;  6312          7823  4         MAP_POINTER = .HEADER + .HEADER[FH1$B_MPOFFSET] * 2;
;  6313          7824  4         EXT_FILE_ID[FID$W_NUM] = .MAP_POINTER[FM1$W_EX_FILNUM];
;  6314          7825  4         EXT_FILE_ID[FID$W_SEQ] = .MAP_POINTER[FM1$W_EX_FILSEQ];
;  6315          7826  4         EXT_FILE_ID[FID$W_RVN] = 0;
;  6316          7827  3         END;
;  6317          7828  3     IF .CURRENT_MTL[MTL_SEQ_DISK]
;  6318          7829  4     OR (.EXT_FILE_ID[FID$W_NUM] EQL 0 AND .EXT_FILE_ID[FID$W_RVN] EQL 0)
;  6319          7830  4     OR (.EXT_FILE_ID[FID$B_RVN] NEQ 0 AND .QUA[QUA_VOLU])
;  6320          7831  3     THEN EXITLOOP;
```

```
;  6321          7832  3     IF .EXT_FILE_ID[FID$B_RVN] EQL 0 THEN EXT_FILE_ID[FID$B_RVN] = .RVN;
;  6322          7833  3
;  6323          7834  3
;  6324          7835  3 ! Set up header and RVN for next trip through loop.
;  6325          7836  3 !
;  6326          7837  3
;  6327          7838  3     FID = EXT_FILE_ID;
;  6328          7839  3     HEADER = LOCAL_HEADER;
;  6329          7840  3     RVN = .EXT_FILE_ID[FID$B_RVN];
;  6330          7841  3
;  6331          7842  3 ! Read extension file header.  If this fails,
;  6332          7843  3 ! exit the loop.
;  6333          7844  3 !
;  6334          7845  3     STATUS = READ_HEADER(EXT_FILE_ID, .HEADER);
;  6335          7846  3     IF NOT .STATUS
;  6336          7847  3     THEN RETURN .STATUS;
;  6337          7848  3     END;
;  6338          7849  3
;  6339          7850  3
;  6340          7851  3 ! Completed normally.
;  6341          7852  3 !
;  6342          7853  3 SS$_NORMAL
;  6343          7854  1 END;


                                03FC 00000 STA_MODIFY:
                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                    ;  7719
                59 00000000'    EF  9E 00002            MOVAB   CURRENT_MTL, R9
                5E       FDF8   CE  9E 00009            MOVAB   -520(SP), SP
                55         1C   AC  D0 0000E            MOVL    P1, R5                                  ;  7762
                                04  13 00012            BEQL    1$
                54         04   A5  D0 00014            MOVL    4(R5), FIB
                50              69  D0 00018 1$:        MOVL    CURRENT_MTL, R0                         ;  7766
                53         0C   A0  D0 0001B            MOVL    12(R0), HEADER
                52         18   A0  9E 0001F            MOVAB   24(R0), FID                             ;  7767
                           08   A0  D5 00023            TSTL    8(R0)                                   ;  7768
                                25  12 00026            BNEQ    3$
                                55  D5 00028            TSTL    R5                                      ;  7771
                                04  12 0002A            BNEQ    2$
                50              14  D0 0002C            MOVL    #20, R0
                                04  0002F              RET
                62         04   A4  D0 00030 2$:        MOVL    4(FIB), (FID)                           ;  7772
              04 A2        08   A4  B0 00034            MOVW    8(FIB), 4(FID)                          ;  7774
                                0C  BB 00039            PUSHR   #^M<R2,R3>                              ;  7775
            CBD2 CF             02  FB 0003B            CALLS   #2, READ_HEADER
                57              50  D0 00040            MOVL    R0, STATUS
                30              57  E9 00043            BLBC    STATUS, 7$                              ;  7776
                50              69  D0 00046            MOVL    CURRENT_MTL, R0                         ;  7777
              31 A0             02  8A 00049            BICB2   #2, 49(R0)
                58         04   A2  9A 0004D 3$:        MOVZBL  4(FID), RVN                             ;  7779
                                55  D5 00051 4$:        TSTL    R5                                      ;  7791
                                04  12 00053            BNEQ    5$
                                7E  D4 00055            CLRL    -(SP)
                                03  11 00057            BRB     6$
```

```
                                04   A5   DD 00059 5$:     PUSHL   4(R5)                                   ;
                                2C   AC   DD 0005C 6$:     PUSHL   P5                                      ;
                                     53   DD 0005F         PUSHL   HEADER                                  ;
                        0000V   CF        03   FB 00061    CALLS   #3, WRITE_ATTRIBUTES                    ;
                                57        50   D0 00066    MOVL    R0, STATUS                              ;
                                0A        57   E9 00069    BLBC    STATUS, 7$                              ; 7792
                                0C        BB 0006C         PUSHR   #^M<R2,R3>                              ; 7797
                        CC2F    CF        02   FB 0006E    CALLS   #2, WRITE_HEADER                        ;
                                57        50   D0 00073    MOVL    R0, STATUS                              ;
                                03        57   E8 00076 7$: BLBS    STATUS, 8$                              ; 7798
                                        0087   31 00079    BRW     16$                                     ;
                                     56   D4 0007C 8$:     CLRL    R6                                      ; 7804
                                02   07   A3 91 0007E      CMPB    7(HEADER), #2                           ;
                                     21   12 00082         BNEQ    10$                                     ;
                                     56   D6 00084         INCL    R6                                      ;
                                1A F996   C9 E8 00086      BLBS    QUAL+14, 10$                            ;
                                     30   AC D5 0008B      TSTL    P6                                      ; 7805
                                     15   13 0008E         BEQL    10$                                     ;
                                04   04   A4 B1 00090      CMPW    4(FIB), #4                              ; 7806
                                     05   1E 00094         BGEQU   9$                                      ;
                                     09   A4 95 00096      TSTB    9(FIB)                                  ;
                                     0A   13 00099         BEQL    10$                                     ;
                                     01   DD 0009B 9$:     PUSHL   #1                                      ; 7808
                                     3C   A3 DD 0009D      PUSHL   60(HEADER)                              ;
                        C90A    CF        02   FB 000A0    CALLS   #2, DQF_MODIFY_USAGE                    ;
                                0C        56   E9 000A5 10$: BLBC    R6, 11$                                 ; 7813
                          F8    AD   0E   A3 D0 000A8      MOVL    14(HEADER), EXT_FILE_ID                 ; 7816
                          FC    AD   12   A3 B0 000AD      MOVW    18(HEADER), EXT_FILE_ID+4               ; 7818
                                     10   11 000B2         BRB     12$                                     ; 7813
                                50   01   A3 9A 000B4 11$: MOVZBL  1(HEADER), R0                          ; 7823
                                50        6340   3E 000B8  MOVAW   (HEADER)[R0], MAP_POINTER               ;
                          F8    AD   02   A0 D0 000BC      MOVL    2(MAP_POINTER), EXT_FILE_ID             ; 7824
                                FC   AD   B4 000C1         CLRW    EXT_FILE_ID+4                           ; 7826
                                50        69   D0 000C4 12$: MOVL    CURRENT_MTL, R0                         ; 7828
                                3C   31   A0 E8 000C7      BLBS    49(R0), 17$                             ;
                                     F8   AD B5 000CB      TSTW    EXT_FILE_ID                             ; 7829
                                     05   12 000CE         BNEQ    13$                                     ;
                                     FC   AD B5 000D0      TSTW    EXT_FILE_ID+4                           ;
                                     32   13 000D3         BEQL    17$                                     ;
                                     FC   AD 95 000D5 13$: TSTB    EXT_FILE_ID+4                           ; 7830
                                     07   13 000D8         BEQL    14$                                     ;
                                28 F996   C9 E8 000DA      BLBS    QUAL+14, 17$                            ;
                                     04   12 000DF         BNEQ    15$                                     ; 7832
                          FC    AD        58   90 000E1 14$: MOVB    RVN, EXT_FILE_ID+4                      ;
                                52   F8   AD 9E 000E5 15$: MOVAB   EXT_FILE_ID, FID                       ; 7838
                                53        6E   9E 000E9    MOVAB   LOCAL_HEADER, HEADER                    ; 7839
                                58   FC   AD 9A 000EC      MOVZBL  EXT_FILE_ID+4, RVN                      ; 7840
                                     53   DD 000F0         PUSHL   HEADER                                  ; 7845
                                     F8   AD 9F 000F2      PUSHAB  EXT_FILE_ID                             ;
                        CB18    CF        02   FB 000F5    CALLS   #2, READ_HEADER                         ;
                                57        50   D0 000FA    MOVL    R0, STATUS                              ;
                                03        57   E9 000FD    BLBC    STATUS, 16$                             ; 7846
                                        FF4E   31 00100    BRW     4$                                      ;
                                50        57   D0 00103 16$: MOVL    STATUS, R0                              ; 7847
                                          04   00106       RET                                             ;
                                50        01   D0 00107 17$: MOVL    #1, R0                                  ; 7854
                                          04   0010A       RET                                             ;
```

; Routine Size: 267 bytes,   Routine Base: CODE + 369E

```
;  6345          7855  1 %SBTTL 'STA_QIO - stand-alone QIO dispatcher'
;  6346          7856  1 GLOBAL ROUTINE STA_QIO (EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)=
;  6347          7857  1
;  6348          7858  1 !++
;  6349          7859  1 !
;  6350          7860  1 ! FUNCTIONAL DESCRIPTION:
;  6351          7861  1 !       This routine is the dispatcher for standalone ACP QIO functions.
;  6352          7862  1 !
;  6353          7863  1 ! INPUT PARAMETERS:
;  6354          7864  1 !       As for $QIO(W) system service.
;  6355          7865  1 !
;  6356          7866  1 ! IMPLICIT INPUTS:
;  6357          7867  1 !       INPUT_MTL       - Pointer to MTL for input volume set.
;  6358          7868  1 !       OUTPUT_MTL      - Pointer to MTL for output volume set.
;  6359          7869  1 !
;  6360          7870  1 ! OUTPUT PARAMETERS:
;  6361          7871  1 !       NONE
;  6362          7872  1 !
;  6363          7873  1 ! IMPLICIT OUTPUTS:
;  6364          7874  1 !       NONE
;  6365          7875  1 !
;  6366          7876  1 ! ROUTINE VALUE:
;  6367          7877  1 !       Completion status.
;  6368          7878  1 !
;  6369          7879  1 ! SIDE EFFECTS:
;  6370          7880  1 !       NONE
;  6371          7881  1 !
;  6372          7882  1 !--
;  6373          7883  1
;  6374          7884  2 BEGIN
;  6375          7885  2 MAP
;  6376          7886  2         FUNC:           BBLOCK,         ! I/O function code
;  6377          7887  2         IOSB:           REF VECTOR [,WORD]; ! I/O status block
;  6378          7888  2 LOCAL
;  6379          7889  2         STATUS;
;  6380          7890  2 BUILTIN
;  6381          7891  2         AP;
;  6382          7892  2
;  6383          7893  2
;  6384          7894  2 IF .IOSB NEQ 0
;  6385          7895  2 THEN
;  6386          7896  3     BEGIN
;  6387          7897  3     IOSB[0] = 0;
;  6388          7898  3     IOSB[1] = 0;
;  6389          7899  3     IOSB[2] = 0;
;  6390          7900  3     IOSB[3] = 0;
;  6391          7901  2     END;
;  6392          7902  2
;  6393          7903  2
;  6394          7904  2 ! Validate the channel number.  The standalone ACP uses special channel numbers
;  6395          7905  2 ! larger than 16 bits to avoid interacting with VMS channel numbers, which are
;  6396          7906  2 ! invalid here.  Set CURRENT_MTL to point to the MTL for the selected volume
;  6397          7907  2 ! set.
;  6398          7908  2 !
;  6399          7909  2 IF
;  6400          7910  3     BEGIN
;  6401          7911  3     SELECTONE .CHAN OF
```

```
; 6402          7912  3          SET
; 6403          7913  3
; 6404          7914  3          [STA_IN_CHAN]:
; 6405          7915  4              BEGIN
; 6406          7916  4              CURRENT_MTL = .INPUT_MTL;
; 6407          7917  4              .CURRENT_MTL EQL 0
; 6408          7918  3              END;
; 6409          7919  3
; 6410          7920  3          [STA_OUT_CHAN]:
; 6411          7921  4              BEGIN
; 6412          7922  4              CURRENT_MTL = .OUTPUT_MTL;
; 6413          7923  4              .CURRENT_MTL EQL 0
; 6414          7924  3              END;
; 6415          7925  3
; 6416          7926  3          [OTHERWISE]:
; 6417          7927  3              TRUE;
; 6418          7928  3
; 6419          7929  3          TES
; 6420          7930  3      END
; 6421          7931  2  THEN
; 6422          7932  2      RETURN SS$_IVCHAN;
; 6423          7933  2
; 6424          7934  2
; 6425          7935  2  ! Dispatch to the function code specific processing routine.
; 6426          7936  2  !
; 6427          7937  2  CASE .FUNC[IO$V_FCODE] FROM IO$_WRITEVBLK TO IO$_MODIFY OF
; 6428          7938  2      SET
; 6429          7939  2
; 6430          7940  2      [IO$_WRITEVBLK, IO$_READVBLK]:
; 6431          7941  3          BEGIN
; 6432          7942  3          STATUS = CALLG(.AP, STA_RDWRVBLK);
; 6433          7943  3          IF .STATUS THEN RETURN .STATUS;
; 6434          7944  2          END;
; 6435          7945  2
; 6436          7946  2      [IO$_ACCESS]:
; 6437          7947  2          STATUS = CALLG(.AP, STA_ACCESS);
; 6438          7948  2
; 6439          7949  2      [IO$_CREATE]:
; 6440          7950  2          STATUS = CALLG(.AP, STA_CREATE);
; 6441          7951  2
; 6442          7952  2      [IO$_DEACCESS]:
; 6443          7953  2          STATUS = CALLG(.AP, STA_DEACCESS);
; 6444          7954  2
; 6445          7955  2      [IO$_MODIFY]:
; 6446          7956  2          STATUS = CALLG(.AP, STA_MODIFY);
; 6447          7957  2
; 6448          7958  2      [INRANGE, OUTRANGE]:
; 6449          7959  2          STATUS = SS$_ILLIOFUNC;
; 6450          7960  2
; 6451          7961  2      TES;
; 6452          7962  2
; 6453          7963  2
; 6454          7964  2  ! Return status in IOSB.
; 6455          7965  2  !
; 6456          7966  2  IF .IOSB NEQ 0
; 6457          7967  2  THEN
; 6458          7968  3      BEGIN
```

```
;  6459          7969  3     IOSB[0] = .STATUS<0,16>;
;  6460          7970  2     END;
;  6461          7971  2
;  6462          7972  2
;  6463          7973  2 ! Set the specified event flag and return success.
;  6464          7974  2 !
;  6465          7975  2 $SETEF(EFN=.EFN);
;  6466          7976  2 SS$_NORMAL
;  6467          7977  1 END;


                                                001C 00000          .ENTRY  STA_QIO, Save R2,R3,R4                  ; 7856
                                    54 00000000'  EF   9E 00002          MOVAB   CURRENT_MTL, R4                         ;
                                    50        10  AC   D0 00009          MOVL    IOSB, R0                                ; 7894
                                                  02   13 0000D          BEQL    1$                                      ;
                                                  60   7C 0000F          CLRQ    (R0)                                    ; 7897
                                    50        08  AC   D0 00011 1$:      MOVL    CHAN, R0                                ; 7911
                                0001FFFF    8F        50   D1 00015          CMPL    R0, #131071                             ; 7914
                                                  06   12 0001C          BNEQ    2$                                      ;
                                    64        F8  A4   D0 0001E          MOVL    INPUT_MTL, CURRENT_MTL                  ; 7916
                                                  0D   11 00022          BRB     3$                                      ; 7917
                                0002FFFF    8F        50   D1 00024 2$:      CMPL    R0, #196607                             ; 7920
                                                  06   12 0002B          BNEQ    4$                                      ;
                                    64        FC  A4   D0 0002D          MOVL    OUTPUT_MTL, CURRENT_MTL                 ; 7922
                                                  06   12 00031 3$:      BNEQ    5$                                      ; 7923
                                    50      013C  8F   3C 00033 4$:      MOVZWL  #316, R0                                ; 7932
                                                        04 00038          RET                                             ;
                53        0C    AC              06   00   EF 00039 5$:      EXTZV   #0, #6, FUNC, R3                        ; 7937
                                    06            30   53   CF 0003F          CASEL   R3, #48, #6                             ;
                002A              0023          0014   0014      00043 6$:      .WORD   8$-6$,-                                 ;
                                  0038          000E   0031      0004B                  8$-6$,-                                 ;
                                                                                      9$-6$,-                                 ;
                                                                                      10$-6$,-                                ;
                                                                                      11$-6$,-                                ;
                                                                                      7$-6$,-                                 ;
                                                                                      12$-6$                                  ;
                                    52        F4  8F   9A 00051 7$:      MOVZBL  #244, STATUS                            ; 7959
                                                  2C   11 00055          BRB     14$                                     ;
                            F2CB    CF            6C   FA 00057 8$:      CALLG   (AP), STA_RDWRVBLK                      ; 7942
                                    52            50   D0 0005C          MOVL    R0, STATUS                              ;
                                    21            52   E9 0005F          BLBC    STATUS, 14$                             ; 7943
                                    50            52   D0 00062          MOVL    STATUS, R0                              ;
                                                        04 00065          RET                                             ;
                            F2DA    CF            6C   FA 00066 9$:      CALLG   (AP), STA_ACCESS                        ; 7947
                                                  13   11 0006B          BRB     13$                                     ;
                            F5F3    CF            6C   FA 0006D 10$:     CALLG   (AP), STA_CREATE                        ; 7950
                                                  0C   11 00072          BRB     13$                                     ;
                            FBBE    CF            6C   FA 00074 11$:     CALLG   (AP), STA_DEACCESS                      ; 7953
                                                  05   11 00079          BRB     13$                                     ;
                            FE75    CF            6C   FA 0007B 12$:     CALLG   (AP), STA_MODIFY                        ; 7956
                                    52            50   D0 00080 13$:     MOVL    R0, STATUS                              ;
                                              10  AC   D5 00083 14$:     TSTL    IOSB                                    ; 7966
                                                  04   13 00086          BEQL    15$                                     ;
                              10    BC            52   B0 00088          MOVW    STATUS, @IOSB                           ; 7969
```

```
                                    04   AC  DD 0008C 15$:    PUSHL   EFN                                     ; 7975
                                         01  FB 0008F        CALLS   #1, SYS$SETEF                           ;
                                         01  D0 00096        MOVL    #1, R0                                  ; 7977
                                             04 00099        RET                                             ;

; Routine Size:  154 bytes,    Routine Base:  CODE + 37A9
```

```
;  6469          7978  1 %SBTTL 'STA_QIOW - stand-alone QIOW dispatcher'
;  6470          7979  1 GLOBAL ROUTINE STA_QIOW (EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)=
;  6471          7980  1
;  6472          7981  1 !++
;  6473          7982  1 !
;  6474          7983  1 ! FUNCTIONAL DESCRIPTION:
;  6475          7984  1 !       This routine executes the $QIOW service for standalone functions.
;  6476          7985  1 !
;  6477          7986  1 ! INPUT PARAMETERS:
;  6478          7987  1 !       As for $QIOW system service.
;  6479          7988  1 !
;  6480          7989  1 ! IMPLICIT INPUTS:
;  6481          7990  1 !       NONE
;  6482          7991  1 !
;  6483          7992  1 ! OUTPUT PARAMETERS:
;  6484          7993  1 !       NONE
;  6485          7994  1 !
;  6486          7995  1 ! IMPLICIT OUTPUTS:
;  6487          7996  1 !       NONE
;  6488          7997  1 !
;  6489          7998  1 ! ROUTINE VALUE:
;  6490          7999  1 !       Completion status.
;  6491          8000  1 !
;  6492          8001  1 ! SIDE EFFECTS:
;  6493          8002  1 !       NONE
;  6494          8003  1 !
;  6495          8004  1 !--
;  6496          8005  1
;  6497          8006  2 BEGIN
;  6498          8007  2 LOCAL
;  6499          8008  2         STATUS;
;  6500          8009  2 BUILTIN
;  6501          8010  2         AP;
;  6502          8011  2
;  6503          8012  2
;  6504          8013  2 STATUS = CALLG(.AP, STA_QIO);
;  6505          8014  2 IF NOT .STATUS THEN RETURN .STATUS;
;  6506          8015  3 $WAITFR(EFN=.EFN)
;  6507          8016  1 END;


                                        0000 00000          .ENTRY  STA_QIOW, Save nothing                  ; 7979
                            FF5F    CF              6C  FA 00002        CALLG   (AP), STA_QIO                   ; 8013
                                    0A              50  E9 00007        BLBC    STATUS, 1$                      ; 8014
                                                04  AC  DD 0000A        PUSHL   EFN                             ; 8015
                        00000000G   00              01  FB 0000D        CALLS   #1, SYS$WAITFR                  ;
                                                    04 00014 1$:        RET                                     ; 8016

; Routine Size:  21 bytes,    Routine Base:  CODE + 3843
```

```
; 6509          8017  1 %SBTTL 'READ_ATTRIBUTES - read file attributes'
; 6510          8018  1 ROUTINE READ_ATTRIBUTES (HEADER,ATRLIST,FIB)=
; 6511          8019  1
; 6512          8020  1 !++
; 6513          8021  1 !
; 6514          8022  1 ! FUNCTIONAL DESCRIPTION:
; 6515          8023  1 !       This routine processes an ACP attributes list, returning the specified
; 6516          8024  1 !       attributes to the user.  Only attributes that are required by BACKUP
; 6517          8025  1 !       are processed.
; 6518          8026  1 !
; 6519          8027  1 ! INPUT PARAMETERS:
; 6520          8028  1 !       HEADER          - Pointer to file header.
; 6521          8029  1 !       ATRLIST         - Pointer to ACP attributes list.
; 6522          8030  1 !       FIB             - Address of the FIB
; 6523          8031  1 !
; 6524          8032  1 ! IMPLICIT INPUTS:
; 6525          8033  1 !       CURRENT_MTL     - Pointer to MTL for selected volume set.
; 6526          8034  1 !
; 6527          8035  1 ! OUTPUT PARAMETERS:
; 6528          8036  1 !       NONE
; 6529          8037  1 !
; 6530          8038  1 ! IMPLICIT OUTPUTS:
; 6531          8039  1 !       NONE
; 6532          8040  1 !
; 6533          8041  1 ! ROUTINE VALUE:
; 6534          8042  1 !       Completion status, SS$_NORMAL or SS$_BADATTRIB.
; 6535          8043  1 !
; 6536          8044  1 ! SIDE EFFECTS:
; 6537          8045  1 !       NONE
; 6538          8046  1 !
; 6539          8047  1 !--
; 6540          8048  1
; 6541          8049  2 BEGIN
; 6542          8050  2 MAP
; 6543          8051  2         HEADER:         REF BBLOCK;             ! Pointer to file header
; 6544          8052  2 LOCAL
; 6545          8053  2         ATR:            REF BBLOCK;             ! Pointer to attribute list
; 6546          8054  2
; 6547          8055  2
; 6548          8056  2 ATR = .ATRLIST;
; 6549          8057  2 WHILE .ATR[ATR$W_TYPE] NEQ 0 DO
; 6550          8058  3     BEGIN
; 6551          8059  3     LOCAL
; 6552          8060  3         S_LEN,                                  ! Source length for move
; 6553          8061  3         S_ADR,                                  ! Source address for move
; 6554          8062  3         TEMP_AREA:      BBLOCK[10],             ! Temporary buffer
; 6555          8063  3         ACL_ATR;                                ! True if an ACL attribute
; 6556          8064  3
; 6557          8065  3
; 6558          8066  3     ACL_ATR = FALSE;
; 6559          8067  3     SELECTONE .ATR[ATR$W_TYPE] OF
; 6560          8068  3         SET
; 6561          8069  3
; 6562          8070  3
; 6563          8071  3         [ATR$C_RECATTR]:
; 6564          8072  4             BEGIN
; 6565          8073  4             S_LEN = FAT$C_LENGTH;
```

```
; 6566          8074  4             S_ADR = HEADER[FH2$W_RECATTR];
; 6567          8075  4             IF .HEADER[FH2$B_STRUCLEV] EQL 1
; 6568          8076  4             THEN
; 6569          8077  4                 S_ADR = HEADER[FH1$W_RECATTR];
; 6570          8078  3             END;
; 6571          8079  3
; 6572          8080  3
; 6573          8081  3         [ATR$C_STATBLK]:
; 6574          8082  4             BEGIN
; 6575          8083  4             LOCAL
; 6576          8084  4                 WCB:    REF BBLOCK;
; 6577          8085  4
; 6578          8086  4             S_LEN = 8;
; 6579          8087  4             S_ADR = TEMP_AREA;
; 6580          8088  4             TEMP_AREA[0,0,32,0] = 0;
; 6581          8089  4             WCB = .CURRENT_MTL[MTL_WINDOW];
; 6582          8090  4             IF .WCB EQL 0 THEN RETURN SS$_BADATTRIB;
; 6583          8091  4             IF .WCB[WCB_LINK] EQL 0 AND .WCB[WCB_SIZE] EQL 1
; 6584          8092  4             THEN
; 6585          8093  4                 TEMP_AREA[0,0,32,0] =
; 6586          8094  4                     ROT(.BBLOCK[WCB[WCB_S_HEADER,0,0,0], WCB_LBN], 16);
; 6587          8095  4             TEMP_AREA[4,0,32,0] = ROT(.CURRENT_MTL[MTL_FILESIZE], 16);
; 6588          8096  3             END;
; 6589          8097  3
; 6590          8098  3
; 6591          8099  3         [ATR$C_HEADER]:
; 6592          8100  4             BEGIN
; 6593          8101  4             S_LEN = 512;
; 6594          8102  4             S_ADR = .HEADER;
; 6595          8103  3             END;
; 6596          8104  3
; 6597          8105  3
; 6598          8106  3         [ATR$C_UIC]:
; 6599          8107  4             BEGIN
; 6600          8108  4             S_LEN = 4;
; 6601          8109  4             S_ADR = HEADER[FH2$L_FILEOWNER];
; 6602          8110  4             IF .HEADER[FH2$B_STRUCLEV] EQL 1
; 6603          8111  4             THEN
; 6604          8112  5                 BEGIN
; 6605          8113  5                 TEMP_AREA<0,16> = .HEADER[FH1$B_UICMEMBER];
; 6606          8114  5                 TEMP_AREA<16,16> = .HEADER[FH1$B_UICGROUP];
; 6607          8115  5                 S_ADR = TEMP_AREA;
; 6608          8116  4                 END;
; 6609          8117  3             END;
; 6610          8118  3
; 6611          8119  3
; 6612          8120  3         [ATR$C_UCHAR]:
; 6613          8121  4             BEGIN
; 6614          8122  4             S_LEN = 4;
; 6615          8123  4             S_ADR = HEADER[FH2$L_FILECHAR];
; 6616          8124  4             IF .HEADER[FH2$B_STRUCLEV] EQL 1
; 6617          8125  4             THEN
; 6618          8126  5                 BEGIN
; 6619          8127  5                 TEMP_AREA<0,32> = .HEADER[FH1$W_FILECHAR];
; 6620          8128  5                 S_ADR = TEMP_AREA;
; 6621          8129  4                 END;
; 6622          8130  3             END;
```

```
;  6623          8131  3
;  6624          8132  3
;  6625          8133  3          [ATR$C_EXTFID]:
;  6626          8134  4              BEGIN
;  6627          8135  4              IF .HEADER[FH2$B_STRUCLEV] NEQ 2
;  6628          8136  4              THEN RETURN SS$_BADATTRIB;
;  6629          8137  4              S_LEN = 6;
;  6630          8138  4              S_ADR = HEADER[FH2$W_EXT_FID];
;  6631          8139  3              END;
;  6632          8140  3
;  6633          8141  3
;  6634          8142  3          [ATR$C_SEGNUM]:
;  6635          8143  4              BEGIN
;  6636          8144  4              IF .HEADER[FH2$B_STRUCLEV] NEQ 2
;  6637          8145  4              THEN RETURN SS$_BADATTRIB;
;  6638          8146  4              S_LEN = 2;
;  6639          8147  4              S_ADR = HEADER[FH2$W_SEG_NUM];
;  6640          8148  3              END;
;  6641          8149  3
;  6642          8150  3          [ATR$C_ADDACLENT, ATR$C_DELETEACL,
;  6643          8151  3           ATR$C_READACL, ATR$C_ACLLENGTH]:
;  6644          8152  3              IF .HEADER[FH2$B_STRUCLEV] EQL 2
;  6645          8153  3              THEN
;  6646          8154  4                  BEGIN
;  6647          8155  4                  ACL_DISPATCH (.ATR[ATR$W_TYPE], .ATR[ATR$W_SIZE],
;  6648          8156  4                                .ATR[ATR$L_ADDR], .FIB);
;  6649          8157  4                  ACL_ATR = TRUE;
;  6650          8158  4                  END
;  6651          8159  3              ELSE S_LEN = 0;
;  6652          8160  3
;  6653          8161  3
;  6654          8162  3              [OTHERWISE]:
;  6655          8163  3                  RETURN SS$_BADATTRIB;
;  6656          8164  3
;  6657          8165  3
;  6658          8166  3              TES;
;  6659          8167  3
;  6660          8168  3
;  6661          8169  3          IF NOT .ACL_ATR
;  6662          8170  3          THEN CH$COPY(.S_LEN, .S_ADR, 0, .ATR[ATR$W_SIZE], .ATR[ATR$L_ADDR]);
;  6663          8171  3          ATR = .ATR + 8;
;  6664          8172  2          END;
;  6665          8173  2
;  6666          8174  2
;  6667          8175  2 SS$_NORMAL
;  6668          8176  1 END;


                                 007C 00000 READ_ATTRIBUTES:
                                                .WORD   Save R2,R3,R4,R5,R6                     ; 8018
                              5E          0C  C2 00002          SUBL2   #12, SP                   ;
                              56      08  AC  D0 00005          MOVL    ATRLIST, ATR              ; 8056
                              51      02  A6  3C 00009 1$:      MOVZWL  2(ATR), R1                ; 8057
                                          03  12 0000D          BNEQ    2$                        ;
```

```
                                0117  31 0000F          BRW     21$                                     ;
                                  55  D4 00012 2$:      CLRL    ACL_ATR                                 ; 8066
                          04      51  B1 00014          CMPW    R1, #4                                  ; 8071
                                  14  12 00017          BNEQ    3$                                      ;
                          54      20  D0 00019          MOVL    #32, S_LEN                              ; 8073
                          50  04  AC  D0 0001C          MOVL    HEADER, R0                              ; 8074
                          52  14  A0  9E 00020          MOVAB   20(R0), S_ADR                           ;
                          01  07  A0  91 00024          CMPB    7(R0), #1                               ; 8075
                                  7C  12 00028          BNEQ    9$                                      ;
                                0096  31 0002A          BRW     12$                                     ; 8077
                          09      51  B1 0002D 3$:      CMPW    R1, #9                                  ; 8081
                                  2F  12 00030          BNEQ    6$                                      ;
                          54      08  D0 00032          MOVL    #8, S_LEN                               ; 8086
                          52      6E  9E 00035          MOVAB   TEMP_AREA, S_ADR                        ; 8087
                                  6E  D4 00038          CLRL    TEMP_AREA                               ; 8088
                          53 00000000'  EF  D0 0003A    MOVL    CURRENT_MTL, R3                         ; 8089
                          50      08  A3  D0 00041      MOVL    8(R3), WCB                              ;
                                  03  12 00045          BNEQ    4$                                      ; 8090
                                00CB  31 00047          BRW     18$                                     ;
                                  60  D5 0004A 4$:      TSTL    (WCB)                                   ; 8091
                                  0B  12 0004C          BNEQ    5$                                      ;
                          01      08  A0  91 0004E      CMPB    8(WCB), #1                              ;
                                  05  12 00052          BNEQ    5$                                      ;
                6E        18    A0      10  9C 00054    ROTL    #16, 24(WCB), TEMP_AREA                 ; 8094
          04    AE        20    A3      10  9C 00059 5$: ROTL    #16, 32(R3), TEMP_AREA+4                ; 8095
                                  7E  11 0005F          BRB     14$                                     ; 8067
                          0A      51  B1 00061 6$:      CMPW    R1, #10                                 ; 8099
                                  0B  12 00064          BNEQ    7$                                      ;
                          54    0200  8F  3C 00066      MOVZWL  #512, S_LEN                             ; 8101
                          52      04  AC  D0 0006B      MOVL    HEADER, S_ADR                           ; 8102
                                  6E  11 0006F          BRB     14$                                     ; 8067
                          15      51  B1 00071 7$:      CMPW    R1, #21                                 ; 8106
                                  1C  12 00074          BNEQ    8$                                      ;
                          54      04  D0 00076          MOVL    #4, S_LEN                               ; 8108
                          50  04  AC  D0 00079          MOVL    HEADER, R0                              ; 8109
                          52  3C  A0  9E 0007D          MOVAB   60(R0), S_ADR                           ;
                          01  07  A0  91 00081          CMPB    7(R0), #1                               ; 8110
                                  58  12 00085          BNEQ    14$                                     ;
                          6E  08  A0  9B 00087          MOVZBW  8(R0), TEMP_AREA                        ; 8113
          02              AE  09  A0  9B 0008B          MOVZBW  9(R0), TEMP_AREA+2                      ; 8114
                                  1A  11 00090          BRB     10$                                     ; 8115
                          03      51  B1 00092 8$:      CMPW    R1, #3                                  ; 8120
                                  1A  12 00095          BNEQ    11$                                     ;
                          54      04  D0 00097          MOVL    #4, S_LEN                               ; 8122
                          50  04  AC  D0 0009A          MOVL    HEADER, R0                              ; 8123
                          52  34  A0  9E 0009E          MOVAB   52(R0), S_ADR                           ;
                          01  07  A0  91 000A2          CMPB    7(R0), #1                               ; 8124
                                  71  12 000A6 9$:      BNEQ    19$                                     ;
                          6E  0C  A0  3C 000A8          MOVZWL  12(R0), TEMP_AREA                       ; 8127
                          52      6E  9E 000AC 10$:     MOVAB   TEMP_AREA, S_ADR                        ; 8128
                                  68  11 000AF          BRB     19$                                     ; 8067
                          27      51  B1 000B1 11$:     CMPW    R1, #39                                 ; 8133
                                  13  12 000B4          BNEQ    13$                                     ;
                          50  04  AC  D0 000B6          MOVL    HEADER, R0                              ; 8135
                          02  07  A0  91 000BA          CMPB    7(R0), #2                               ;
                                  55  12 000BE          BNEQ    18$                                     ;
                          54      06  D0 000C0          MOVL    #6, S_LEN                               ; 8137
```

```
                                52      0E   A0  9E 000C3 12$:    MOVAB   14(R0), S_ADR                       ; 8138
                                        50   11  000C7           BRB     19$                                 ; 8067
                                28           51  B1 000C9 13$:    CMPW    R1, #40                             ; 8142
                                        13   12  000CC           BNEQ    15$                                 ;
                                50      04   AC  D0 000CE           MOVL    HEADER, R0                          ; 8144
                                02      07   A0  91 000D2           CMPB    7(R0), #2                           ;
                                        3D   12  000D6           BNEQ    18$                                 ;
                                54           02  D0 000D8           MOVL    #2, S_LEN                           ; 8146
                                52      04   A0  9E 000DB           MOVAB   4(R0), S_ADR                        ; 8147
                                        38   11  000DF 14$:        BRB     19$                                 ; 8067
                                1F           51  B1 000E1 15$:    CMPW    R1, #31                             ; 8150
                                        0A   13  000E4           BEQL    16$                                 ;
                                24           51  B1 000E6           CMPW    R1, #36                             ;
                                        2A   1F  000E9           BLSSU   18$                                 ;
                                26           51  B1 000EB           CMPW    R1, #38                             ;
                                        25   1A  000EE           BGTRU   18$                                 ;
                                50      04   AC  D0 000F0 16$:    MOVL    HEADER, R0                          ; 8152
                                02      07   A0  91 000F4           CMPB    7(R0), #2                           ;
                                        17   12  000F8           BNEQ    17$                                 ;
                                        0C   AC  DD 000FA           PUSHL   FIB                                 ; 8156
                                        04   A6  DD 000FD           PUSHL   4(ATR)                              ;
                                7E           66  3C 00100           MOVZWL  (ATR), -(SP)                        ; 8155
                                             51  DD 00103           PUSHL   R1                                  ;
                      00000000G 00           04  FB 00105           CALLS   #4, ACL_DISPATCH                    ;
                                55           01  D0 0010C           MOVL    #1, ACL_ATR                         ; 8157
                                             08  11 0010F           BRB     19$                                 ; 8152
                                             54  D4 00111 17$:      CLRL    S_LEN                               ; 8159
                                             04  11 00113           BRB     19$                                 ; 8152
                                50           34  D0 00115 18$:    MOVL    #52, R0                             ; 8163
                                                 04 00118           RET                                         ;
                                07           55  E8 00119 19$:    BLBS    ACL_ATR, 20$                        ; 8169
                66              00      62   54  2C 0011C           MOVC5   S_LEN, (S_ADR), #0, (ATR), @4(ATR)  ; 8170
                                        04   B6      00121                                                      ;
                                56           08  C0 00123 20$:    ADDL2   #8, ATR                             ; 8171
                                        FEE0 31  00126           BRW     1$                                  ; 8057
                                50           01  D0 00129 21$:    MOVL    #1, R0                              ; 8176
                                                 04 0012C           RET                                         ;

; Routine Size:  301 bytes,    Routine Base:  CODE + 3858
```

```
;  6670          8177  1 %SBTTL 'TO_ODS1_DATE - format ODS-1 date'
;  6671          8178  1 GLOBAL ROUTINE TO_ODS1_DATE (SRC,DST): NOVALUE=
;  6672          8179  1
;  6673          8180  1 !++
;  6674          8181  1 !
;  6675          8182  1 ! FUNCTIONAL DESCRIPTION:
;  6676          8183  1 !       This routine converts a date in 64-bit format to ODS-1 format.
;  6677          8184  1 !
;  6678          8185  1 ! INPUT PARAMETERS:
;  6679          8186  1 !       SRC             - Address of quadword time value.
;  6680          8187  1 !       DST             - Address of 13-byte output buffer.
;  6681          8188  1 !
;  6682          8189  1 ! IMPLICIT INPUTS:
;  6683          8190  1 !       NONE
;  6684          8191  1 !
;  6685          8192  1 ! OUTPUT PARAMETERS:
;  6686          8193  1 !       NONE
;  6687          8194  1 !
;  6688          8195  1 ! IMPLICIT OUTPUTS:
;  6689          8196  1 !       NONE
;  6690          8197  1 !
;  6691          8198  1 ! ROUTINE VALUE:
;  6692          8199  1 !       NONE
;  6693          8200  1 !
;  6694          8201  1 ! SIDE EFFECTS:
;  6695          8202  1 !       NONE
;  6696          8203  1 !
;  6697          8204  1 !--
;  6698          8205  1
;  6699          8206  2 BEGIN
;  6700          8207  2 MAP
;  6701          8208  2         DST:            REF BBLOCK;             ! Pointer to destination
;  6702          8209  2 LOCAL
;  6703          8210  2         DESC:           VECTOR[2],              ! Descriptor for buffer
;  6704          8211  2         BUFFER:         BBLOCK[23];             ! Buffer for converted time
;  6705          8212  2
;  6706          8213  2
;  6707          8214  2 ! If the time value is 0, return the output area filled with binary zeros.
;  6708          8215  2 !
;  6709          8216  2 IF ..SRC EQL 0
;  6710          8217  2 THEN
;  6711          8218  3     BEGIN
;  6712          8219  3     CH$FILL(0, 13, .DST);
;  6713          8220  3     RETURN;
;  6714          8221  2     END;
;  6715          8222  2
;  6716          8223  2
;  6717          8224  2 ! Convert the value.
;  6718          8225  2 !
;  6719          8226  2 DESC[0] = 23;                               ! Make descriptor
;  6720          8227  2 DESC[1] = BUFFER;
;  6721          8228  2 $ASCTIM(TIMBUF=DESC, TIMADR=.SRC);         ! Convert time value
;  6722          8229  2 DST[0,0,16,0] = .BUFFER[0,0,16,0];         ! Output DD
;  6723          8230  2 DST[2,0,24,0] = .BUFFER[3,0,24,0];         ! Output MMM
;  6724          8231  2 DST[5,0,16,0] = .BUFFER[9,0,16,0];         ! Output YY
;  6725          8232  2 DST[7,0,16,0] = .BUFFER[12,0,16,0];       ! Output HH
;  6726          8233  2 DST[9,0,16,0] = .BUFFER[15,0,16,0];       ! Output MM
```

```
; 6727          8234  2 DST[11,0,16,0] = .BUFFER[18,0,16,0];     ! Output SS
; 6728          8235  1 END;


                                                                .EXTRN  SYS$ASCTIM

                                                    003C 00000          .ENTRY  TO_ODS1_DATE, Save R2,R3,R4,R5          ; 8178
                                          5E        20   C2 00002          SUBL2   #32, SP                                 ;
                                                04  BC   D5 00005          TSTL    @SRC                                    ; 8216
                                                    08   12 00008          BNEQ    1$                                      ;
                OD              00        6E        00   2C 0000A          MOVC5   #0, (SP), #0, #13, @DST                 ; 8219
                                                08  BC       0000F                                                          ;
                                                    04   00011          RET                                             ; 8218
                                    18    AE        17   D0 00012 1$:      MOVL    #23, DESC                               ; 8226
                                    1C    AE        6E   9E 00016          MOVAB   BUFFER, DESC+4                          ; 8227
                                                    7E   D4 0001A          CLRL    -(SP)                                   ; 8228
                                                04  AC   DD 0001C          PUSHL   SRC                                     ;
                                                20  AE   9F 0001F          PUSHAB  DESC                                    ;
                                                    7E   D4 00022          CLRL    -(SP)                                   ;
                            00000000G   00          04   FB 00024          CALLS   #4, SYS$ASCTIM                          ;
                                        50      08  AC   D0 0002B          MOVL    DST, R0                                 ; 8229
                                        60          6E   B0 0002F          MOVW    BUFFER, (R0)                            ;
            02  A0          18         00      03  AE   F0 00032          INSV    BUFFER+3, #0, #24, 2(R0)                ; 8230
                                    05  A0      09  AE   B0 00039          MOVW    BUFFER+9, 5(R0)                         ; 8231
                                    07  A0      0C  AE   B0 0003E          MOVW    BUFFER+12, 7(R0)                        ; 8232
                                    09  A0      0F  AE   B0 00043          MOVW    BUFFER+15, 9(R0)                        ; 8233
                                    0B  A0      12  AE   B0 00048          MOVW    BUFFER+18, 11(R0)                       ; 8234
                                                    04   0004D          RET                                             ; 8235

; Routine Size:  78 bytes,    Routine Base:  CODE + 3985
```

```
; 6730          8236  1 %SBTTL 'WRITE_ATTRIBUTES - write file attributes'
; 6731          8237  1 ROUTINE WRITE_ATTRIBUTES (HEADER,ATR_LIST,FIB)=
; 6732          8238  1
; 6733          8239  1 !++
; 6734          8240  1 !
; 6735          8241  1 ! FUNCTIONAL DESCRIPTION:
; 6736          8242  1 !       This routine processes an ACP attributes list, writing the specified
; 6737          8243  1 !       attributes to the header.  Only attributes that are required by BACKUP
; 6738          8244  1 !       are processed.
; 6739          8245  1 !
; 6740          8246  1 ! INPUT PARAMETERS:
; 6741          8247  1 !       HEADER          - Pointer to file header.
; 6742          8248  1 !       ATR_LIST        - Pointer to ACP attributes list.
; 6743          8249  1 !       FIB             - Address of the FIB
; 6744          8250  1 !
; 6745          8251  1 ! IMPLICIT INPUTS:
; 6746          8252  1 !       NONE
; 6747          8253  1 !
; 6748          8254  1 ! OUTPUT PARAMETERS:
; 6749          8255  1 !       NONE
; 6750          8256  1 !
; 6751          8257  1 ! IMPLICIT OUTPUTS:
; 6752          8258  1 !       NONE
; 6753          8259  1 !
; 6754          8260  1 ! ROUTINE VALUE:
; 6755          8261  1 !       Completion status, SS$_NORMAL or SS$_BADATTRIB.
; 6756          8262  1 !
; 6757          8263  1 ! SIDE EFFECTS:
; 6758          8264  1 !       NONE
; 6759          8265  1 !
; 6760          8266  1 !--
; 6761          8267  1
; 6762          8268  2 BEGIN
; 6763          8269  2 MAP
; 6764          8270  2         HEADER:         REF BBLOCK;             ! Pointer to file header
; 6765          8271  2 LOCAL
; 6766          8272  2         ATR:            REF BBLOCK,             ! Pointer to attribute list
; 6767          8273  2         IDENT_AREA:     REF BBLOCK,             ! Pointer to ident area of header
; 6768          8274  2         IDENT_LEN;                              ! Length of ident area
; 6769          8275  2
; 6770          8276  2
; 6771          8277  2 ATR = .ATR_LIST;
; 6772          8278  2 IDENT_AREA = .HEADER + .HEADER[FH2$B_IDOFFSET] * 2;
; 6773          8279  2 IDENT_LEN = (.HEADER[FH2$B_MPOFFSET] - .HEADER[FH2$B_IDOFFSET]) * 2;
; 6774          8280  2 WHILE .ATR[ATR$W_TYPE] NEQ 0 DO
; 6775          8281  3     BEGIN
; 6776          8282  3     LOCAL
; 6777          8283  3         S_LEN,                                  ! Source length for move
; 6778          8284  3         S_ADR,                                  ! Source address for move
; 6779          8285  3         D_LEN,                                  ! Destination length for move
; 6780          8286  3         D_ADR,                                  ! Destination address for move
; 6781          8287  3         TEMP_AREA:      BBLOCK[13],             ! Temporary buffer
; 6782          8288  3         ACL_ATR;                                ! True if ACL attribute
; 6783          8289  3
; 6784          8290  3
; 6785          8291  3     S_LEN = .ATR[ATR$W_SIZE];
; 6786          8292  3     S_ADR = .ATR[ATR$L_ADDR];
```

```
; 6787          8293  3
; 6788          8294  3
; 6789          8295  3     ACL_ATR = FALSE;
; 6790          8296  3     CASE .ATR[ATR$W_TYPE] FROM ATR$C_UCHAR TO ATR$C_SEGNUM OF
; 6791          8297  3         SET
; 6792          8298  3
; 6793          8299  3
; 6794          8300  3         [ATR$C_UCHAR]:
; 6795          8301  4             BEGIN
; 6796          8302  4             LITERAL
; 6797          8303  4                 PROTECTED_CHAR=
; 6798          8304  4                     FCH$M_CONTIG OR
; 6799          8305  4                     FCH$M_SPOOL OR
; 6800          8306  4                     FCH$M_BADBLOCK OR
; 6801          8307  4                     FCH$M_MARKDEL,
; 6802          8308  4                 ODS1_CHAR=
; 6803          8309  4                     FCH$M_NOBACKUP OR
; 6804          8310  4                     FCH$M_READCHECK OR
; 6805          8311  4                     FCH$M_WRITCHECK OR
; 6806          8312  4                     FCH$M_CONTIGB OR
; 6807          8313  4                     FCH$M_LOCKED OR
; 6808          8314  4                     FCH$M_CONTIG OR
; 6809          8315  4                     FCH$M_SPOOL OR
; 6810          8316  4                     FCH$M_BADBLOCK OR
; 6811          8317  4                     FCH$M_MARKDEL;
; 6812          8318  4
; 6813          8319  4
; 6814          8320  4             D_LEN = 4;
; 6815          8321  4             D_ADR = HEADER[FH2$L_FILECHAR];
; 6816          8322  4             IF .HEADER[FH2$B_STRUCLEV] EQL 1
; 6817          8323  4             THEN
; 6818          8324  5                 BEGIN
; 6819          8325  5                 D_LEN = 2;
; 6820          8326  5                 D_ADR = HEADER[FH1$W_FILECHAR];
; 6821          8327  4                 END;
; 6822          8328  4
; 6823          8329  4
; 6824          8330  4             ! Get value from user attribute list.
; 6825          8331  4             !
; 6826          8332  4             CH$COPY(.S_LEN, .S_ADR, 0, .D_LEN, TEMP_AREA);
; 6827          8333  4
; 6828          8334  4
; 6829          8335  4             ! Get protected characteristics from file header.
; 6830          8336  4             !
; 6831          8337  4             TEMP_AREA =
; 6832          8338  4                 (.TEMP_AREA AND NOT PROTECTED_CHAR) OR
; 6833          8339  4                 (.(.D_ADR)<0,.D_LEN*8> AND PROTECTED_CHAR);
; 6834          8340  4
; 6835          8341  4
; 6836          8342  4             ! Make sure characteristics bits not used in ODS-1 are clear.
; 6837          8343  4             !
; 6838          8344  4             IF .HEADER[FH2$B_STRUCLEV] EQL 1
; 6839          8345  4             THEN
; 6840          8346  4                 TEMP_AREA = .TEMP_AREA AND ODS1_CHAR;
; 6841          8347  4
; 6842          8348  4
; 6843          8349  4             S_LEN = .D_LEN;
```

```
;  6844     8350  4              S_ADR = TEMP_AREA;
;  6845     8351  3              END;
;  6846     8352  3
;  6847     8353  3
;  6848     8354  3          [ATR$C_RECATTR]:
;  6849     8355  4              BEGIN
;  6850     8356  4              D_LEN = FAT$C_LENGTH;
;  6851     8357  4              D_ADR = HEADER[FH2$W_RECATTR];
;  6852     8358  4              IF .HEADER[FH2$B_STRUCLEV] EQL 1
;  6853     8359  4              THEN
;  6854     8360  4                  D_ADR = HEADER[FH1$W_RECATTR];
;  6855     8361  3              END;
;  6856     8362  3
;  6857     8363  3
;  6858     8364  3          [ATR$C_ASCDATES]:
;  6859     8365  4              BEGIN
;  6860     8366  4              D_LEN = 2;
;  6861     8367  4              D_ADR = IDENT_AREA[FI2$W_REVISION];
;  6862     8368  4              IF .HEADER[FH2$B_STRUCLEV] EQL 1
;  6863     8369  4              THEN
;  6864     8370  4                  D_ADR = IDENT_AREA[FI1$W_REVISION]
;  6865     8371  4              ELSE
;  6866     8372  4                  IF .IDENT_LEN LSSU $BYTEOFFSET(FI2$W_REVISION) + 2
;  6867     8373  4                  THEN
;  6868     8374  4                      D_LEN = 0;
;  6869     8375  3              END;
;  6870     8376  3
;  6871     8377  3
;  6872     8378  3          [ATR$C_CREDATE]:
;  6873     8379  4              BEGIN
;  6874     8380  4              D_LEN = 8;
;  6875     8381  4              D_ADR = IDENT_AREA[FI2$Q_CREDATE];
;  6876     8382  4              IF .HEADER[FH2$B_STRUCLEV] EQL 1
;  6877     8383  4              THEN
;  6878     8384  5                  BEGIN
;  6879     8385  5                  D_ADR = IDENT_AREA[FI1$T_CREDATE];
;  6880     8386  5                  TO_ODS1_DATE(.S_ADR, TEMP_AREA);
;  6881     8387  5                  S_LEN = 13;
;  6882     8388  5                  S_ADR = TEMP_AREA;
;  6883     8389  5                  D_LEN = 13;
;  6884     8390  5                  END
;  6885     8391  4              ELSE
;  6886     8392  4                  IF .IDENT_LEN LSSU $BYTEOFFSET(FI2$Q_CREDATE) + 8
;  6887     8393  4                  THEN
;  6888     8394  4                      D_LEN = 0;
;  6889     8395  3              END;
;  6890     8396  3
;  6891     8397  3
;  6892     8398  3          [ATR$C_REVDATE]:
;  6893     8399  4              BEGIN
;  6894     8400  4              D_LEN = 8;
;  6895     8401  4              D_ADR = IDENT_AREA[FI2$Q_REVDATE];
;  6896     8402  4              IF .HEADER[FH2$B_STRUCLEV] EQL 1
;  6897     8403  4              THEN
;  6898     8404  5                  BEGIN
;  6899     8405  5                  D_ADR = IDENT_AREA[FI1$T_REVDATE];
;  6900     8406  5                  TO_ODS1_DATE(.S_ADR, TEMP_AREA);
```

```
; 6901          8407  5                 S_LEN = 13;
; 6902          8408  5                 S_ADR = TEMP_AREA;
; 6903          8409  5                 D_LEN = 13;
; 6904          8410  5                 END
; 6905          8411  4             ELSE
; 6906          8412  4                 IF .IDENT_LEN LSSU $BYTEOFFSET(FI2$Q_REVDATE) + 8
; 6907          8413  4                 THEN
; 6908          8414  4                     D_LEN = 0;
; 6909          8415  3             END;
; 6910          8416  3
; 6911          8417  3
; 6912          8418  3         [ATR$C_EXPDATE]:
; 6913          8419  4             BEGIN
; 6914          8420  4             D_LEN = 8;
; 6915          8421  4             D_ADR = IDENT_AREA[FI2$Q_EXPDATE];
; 6916          8422  4             IF .HEADER[FH2$B_STRUCLEV] EQL 1
; 6917          8423  4             THEN
; 6918          8424  5                 BEGIN
; 6919          8425  5                 D_ADR = IDENT_AREA[FI1$T_EXPDATE];
; 6920          8426  5                 TO_ODS1_DATE(.S_ADR, TEMP_AREA);
; 6921          8427  5                 S_LEN = 13;
; 6922          8428  5                 S_ADR = TEMP_AREA;
; 6923          8429  5                 D_LEN = 7;
; 6924          8430  5                 END
; 6925          8431  4             ELSE
; 6926          8432  4                 IF .IDENT_LEN LSSU $BYTEOFFSET(FI2$Q_EXPDATE) + 8
; 6927          8433  4                 THEN
; 6928          8434  4                     D_LEN = 0;
; 6929          8435  3             END;
; 6930          8436  3
; 6931          8437  3
; 6932          8438  3         [ATR$C_BAKDATE]:
; 6933          8439  4             BEGIN
; 6934          8440  4             D_LEN = 8;
; 6935          8441  4             D_ADR = IDENT_AREA[FI2$Q_BAKDATE];
; 6936          8442  4             IF .HEADER[FH2$B_STRUCLEV] EQL 1
; 6937          8443  4             THEN
; 6938          8444  4                 D_LEN = 0
; 6939          8445  4             ELSE
; 6940          8446  4                 IF .IDENT_LEN LSSU $BYTEOFFSET(FI2$Q_BAKDATE) + 8
; 6941          8447  4                 THEN
; 6942          8448  4                     D_LEN = 0;
; 6943          8449  3             END;
; 6944          8450  3
; 6945          8451  3
; 6946          8452  3         [ATR$C_UIC]:
; 6947          8453  4             BEGIN
; 6948          8454  4             D_LEN = 4;
; 6949          8455  4             D_ADR = HEADER[FH2$L_FILEOWNER];
; 6950          8456  4             IF .HEADER[FH2$B_STRUCLEV] EQL 1
; 6951          8457  4             THEN
; 6952          8458  5                 BEGIN
; 6953          8459  5                 TEMP_AREA<0,8> = .(.S_ADR)<0,16>;
; 6954          8460  5                 TEMP_AREA<8,8> = .(.S_ADR)<16,16>;
; 6955          8461  5                 S_LEN = 2;
; 6956          8462  5                 S_ADR = TEMP_AREA;
; 6957          8463  5                 D_LEN = 2;
```

```
;  6958          8464  5                 D_ADR = HEADER[FH1$W_FILEOWNER];
;  6959          8465  4                 END;
;  6960          8466  3             END;
;  6961          8467  3
;  6962          8468  3
;  6963          8469  3         [ATR$C_FPRO]:
;  6964          8470  4             BEGIN
;  6965          8471  4             D_LEN = 2;
;  6966          8472  4             D_ADR = HEADER[FH2$W_FILEPROT];
;  6967          8473  4             IF .HEADER[FH2$B_STRUCLEV] EQL 1
;  6968          8474  4             THEN
;  6969          8475  4                 D_ADR = HEADER[FH1$W_FILEPROT];
;  6970          8476  3             END;
;  6971          8477  3
;  6972          8478  3
;  6973          8479  3         [ATR$C_RPRO]:
;  6974          8480  4             BEGIN
;  6975          8481  4             D_LEN = 2;
;  6976          8482  4             D_ADR = HEADER[FH2$W_RECPROT];
;  6977          8483  4             IF .HEADER[FH2$B_STRUCLEV] EQL 1
;  6978          8484  4             THEN
;  6979          8485  4                 D_LEN = 0;
;  6980          8486  3             END;
;  6981          8487  3
;  6982          8488  3
;  6983          8489  3         [ATR$C_JOURNAL]:
;  6984          8490  4             BEGIN
;  6985          8491  4             D_LEN = 2;
;  6986          8492  4             D_ADR = HEADER[FH2$W_JOURNAL];
;  6987          8493  4             IF .HEADER[FH2$B_STRUCLEV] EQL 1
;  6988          8494  4             THEN
;  6989          8495  4                 D_LEN = 0;
;  6990          8496  3             END;
;  6991          8497  3
;  6992          8498  3
;  6993          8499  3         [ATR$C_ACLEVEL]:
;  6994          8500  4             BEGIN
;  6995          8501  4             D_LEN = 1;
;  6996          8502  4             D_ADR = HEADER[FH2$B_ACC_MODE];
;  6997          8503  4             IF .HEADER[FH2$B_STRUCLEV] EQL 1
;  6998          8504  4             THEN
;  6999          8505  4                 D_LEN = 0;
;  7000          8506  3             END;
;  7001          8507  3
;  7002          8508  3
;  7003          8509  3         [ATR$C_EXTFID]:
;  7004          8510  4             BEGIN
;  7005          8511  4             IF .HEADER[FH2$B_STRUCLEV] NEQ 2
;  7006          8512  4             THEN RETURN SS$_BADATTRIB;
;  7007          8513  4             D_LEN = 6;
;  7008          8514  4             D_ADR = HEADER[FH2$W_EXT_FID];
;  7009          8515  3             END;
;  7010          8516  3
;  7011          8517  3
;  7012          8518  3         [ATR$C_SEGNUM]:
;  7013          8519  4             BEGIN
;  7014          8520  4             IF .HEADER[FH2$B_STRUCLEV] NEQ 2
```

```
;  7015          8521  4              THEN RETURN SS$_BADATTRIB;
;  7016          8522  4              D_LEN = 2;
;  7017          8523  4              D_ADR = HEADER[FH2$W_SEG_NUM];
;  7018          8524  3              END;
;  7019          8525  3
;  7020          8526  3
;  7021          8527  3          [ATR$C_ADDACLENT, ATR$C_DELETEACL,
;  7022          8528  3           ATR$C_READACL, ATR$C_ACLLENGTH]:
;  7023          8529  4              BEGIN
;  7024          8530  4              IF .HEADER[FH2$B_STRUCLEV] EQL 2
;  7025          8531  4              THEN ACL_DISPATCH (.ATR[ATR$W_TYPE], .ATR[ATR$W_SIZE],
;  7026          8532  4                                 .ATR[ATR$L_ADDR], .FIB);
;  7027          8533  4              ACL_ATR = TRUE;
;  7028          8534  3              END;
;  7029          8535  3
;  7030          8536  3
;  7031          8537  3          [INRANGE, OUTRANGE]:
;  7032          8538  3              RETURN SS$_BADATTRIB;
;  7033          8539  3
;  7034          8540  3
;  7035          8541  3          TES;
;  7036          8542  3
;  7037          8543  3
;  7038          8544  3      IF NOT .ACL_ATR THEN CH$COPY(.S_LEN, .S_ADR, 0, .D_LEN, .D_ADR);
;  7039          8545  3      ATR = .ATR + 8;
;  7040          8546  2      END;
;  7041          8547  2
;  7042          8548  2
;  7043          8549  2 SS$_NORMAL
;  7044          8550  1 END;


                                        0FFC 00000 WRITE_ATTRIBUTES:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 8237
                                 5E          20  C2 00002          SUBL2   #32, SP                       ;
                           08    AE      08    AC  D0 00005          MOVL    ATR_LIST, ATR                 ; 8277
                                 59      04    AC  D0 0000A          MOVL    HEADER, R9                    ; 8278
                                 50          69  9A 0000E          MOVZBL  (R9), R0                      ;
                                 58        6940  3E 00011          MOVAW   (R9)[R0], IDENT_AREA          ;
                                 57      01    A9  9A 00015          MOVZBL  1(R9), R7                     ; 8279
                                 57          50  C2 00019          SUBL2   R0, R7                        ;
                     04    AE    57          01  78 0001C          ASHL    #1, R7, IDENT_LEN             ;
                           50    AE          02  C1 00021 1$:      ADDL3   #2, ATR, R0                   ; 8280
                                 51          60  3C 00026          MOVZWL  (R0), R1                      ;
                                             03  12 00029          BNEQ    2$                            ;
                                           0216  31 0002B          BRW     41$                           ;
                                 5B      08    BE  3C 0002E 2$:      MOVZWL  @ATR, S_LEN                   ; 8291
                           50    AE      08    AC  C1 00032          ADDL3   #4, ATR, R0                   ; 8292
                                 5A          60  D0 00037          MOVL    (R0), S_ADR                   ;
                                             6E  D4 0003A          CLRL    ACL_ATR                       ; 8295
                              25             51  AF 0003C          CASEW   R1, #3, #37                   ; 8296
                01C7          01C7        00A3      004F      00040 3$:      .WORD   4$-3$,-               ;
                01C7          01C7        01C7      01C7      00048          7$-3$,-                       ;
                01C7          00B3        01C7      01C7      00050          34$-3$,-                      ;
```

```
                00E5        00CC        01C7        01C7    00058                   34$-3$,-                                                ;
                017C        0156        0141        0112    00060                   34$-3$,-                                                ;
                01C7        01C7        01A1        018F    00068                   34$-3$,-                                                ;
                01C7        0198        01C7        01C7    00070                   34$-3$,-                                                ;
                01C7        01C7        01C7        01D4    00078                   34$-3$,-                                                ;
                01D4        01D4        01D4        01C7    00080                   34$-3$,-                                                ;
                                        01C1        01B2    00088                   34$-3$,-                                                ;
                                                                                    8$-3$,-                                                 ;
                                                                                    34$-3$,-                                                ;
                                                                                    34$-3$,-                                                ;
                                                                                    34$-3$,-                                                ;
                                                                                    10$-3$,-                                                ;
                                                                                    12$-3$,-                                                ;
                                                                                    16$-3$,-                                                ;
                                                                                    20$-3$,-                                                ;
                                                                                    22$-3$,-                                                ;
                                                                                    24$-3$,-                                                ;
                                                                                    25$-3$,-                                                ;
                                                                                    27$-3$,-                                                ;
                                                                                    34$-3$,-                                                ;
                                                                                    34$-3$,-                                                ;
                                                                                    34$-3$,-                                                ;
                                                                                    34$-3$,-                                                ;
                                                                                    26$-3$,-                                                ;
                                                                                    34$-3$,-                                                ;
                                                                                    37$-3$,-                                                ;
                                                                                    34$-3$,-                                                ;
                                                                                    34$-3$,-                                                ;
                                                                                    34$-3$,-                                                ;
                                                                                    34$-3$,-                                                ;
                                                                                    37$-3$,-                                                ;
                                                                                    37$-3$,-                                                ;
                                                                                    37$-3$,-                                                ;
                                                                                    37$-3$,-                                                ;
                                                                                    31$-3$,-                                                ;
                                                                                    33$-3$                                                  ;
                                                0178  31  0008C         BRW         34$                                                     ; 8538
                                    56          04  D0  0008F  4$:     MOVL        #4, D_LEN                                               ; 8320
                                    57      34  A9  9E  00092         MOVAB       52(R9), D_ADR                                           ; 8321
                                        0C      AE  D4  00096         CLRL        12(SP)                                                  ; 8322
                                    01      07  A9  91  00099         CMPB        7(R9), #1                                               ;
                                                0A  12  0009D         BNEQ        5$                                                      ;
                                        0C      AE  D6  0009F         INCL        12(SP)                                                  ;
                                    56          02  D0  000A2         MOVL        #2, D_LEN                                               ; 8325
                                    57      0C  A9  9E  000A5         MOVAB       12(R9), D_ADR                                           ; 8326
                56          00          6A          5B  2C  000A9  5$:     MOVC5       S_LEN, (S_ADR), #0, D_LEN, TEMP_AREA                    ; 8332
                                            10      AE      000AE                                                                         ;
                        52      10  AE  0000D080    8F  CB  000B0         BICL3       #53376, TEMP_AREA, R2                                   ; 8338
                        51          56              03  78  000B9         ASHL        #3, D_LEN, RT                                           ; 8339
                50      67          51              00  EF  000BD         EXTZV       #0, RT, (D_ADR), R0                                     ;
                                    50  FFFF2F7F    8F  CA  000C2         BICL2       #-53377, R0                                             ;
                    10  AE          50              52  C9  000C9         BISL3       R2, R0, TEMP_AREA                                       ;
                                    08          0C  AE  E9  000CE         BLBC        12(SP), 6$                                              ; 8344
                                10  AE  FFFF2F05    8F  CA  000D2         BICL2       #-53499, TEMP_AREA                                      ; 8346
                                    5B              56  D0  000DA  6$:     MOVL        D_LEN, S_LEN                                            ; 8349
                                    5A          10  AE  9E  000DD         MOVAB       TEMP_AREA, S_ADR                                        ; 8350
                                                67  11  000E1         BRB         14$                                                     ; 8296
                                    56              20  D0  000E3  7$:     MOVL        #32, D_LEN                                              ; 8356
```

```
                                57      14    A9  9E 000E6           MOVAB   20(R9), D_ADR                   ; 8357
                                01      07    A9  91 000EA           CMPB    7(R9), #1                       ; 8358
                                              5A  12 000EE           BNEQ    14$
                                            0108  31 000F0           BRW     32$                             ; 8360
                                56            02  D0 000F3 8$:       MOVL    #2, D_LEN                       ; 8366
                                57      14    A8  9E 000F6           MOVAB   20(R8), D_ADR                   ; 8367
                                01      07    A9  91 000FA           CMPB    7(R9), #1                       ; 8368
                                              06  12 000FE           BNEQ    9$
                                57      0A    A8  9E 00100           MOVAB   10(R8), D_ADR                   ; 8370
                                              73  11 00104           BRB     17$
                                16      04    AE  D1 00106 9$:       CMPL    IDENT_LEN, #22                  ; 8372
                                              73  11 0010A           BRB     19$
                                56            08  D0 0010C 10$:      MOVL    #8, D_LEN                       ; 8380
                                57      16    A8  9E 0010F           MOVAB   22(R8), D_ADR                   ; 8381
                                01      07    A9  91 00113           CMPB    7(R9), #1                       ; 8382
                                              06  12 00117           BNEQ    11$
                                57      19    A8  9E 00119           MOVAB   25(R8), D_ADR                   ; 8385
                                              17  11 0011D           BRB     13$                             ; 8386
                                1E      04    AE  D1 0011F 11$:      CMPL    IDENT_LEN, #30                  ; 8392
                                              6D  11 00123           BRB     21$
                                56            08  D0 00125 12$:      MOVL    #8, D_LEN                       ; 8400
                                57      1E    A8  9E 00128           MOVAB   30(R8), D_ADR                   ; 8401
                                01      07    A9  91 0012C           CMPB    7(R9), #1                       ; 8402
                                              1A  12 00130           BNEQ    15$
                                57      0C    A8  9E 00132           MOVAB   12(R8), D_ADR                   ; 8405
                                        10    AE  9F 00136 13$:      PUSHAB  TEMP_AREA                       ; 8406
                                              5A  DD 00139           PUSHL   S_ADR
                            FE72    CF            02  FB 0013B           CALLS   #2, TO_ODS1_DATE
                                5B            0D  D0 00140           MOVL    #13, S_LEN                      ; 8407
                                5A      10    AE  9E 00143           MOVAB   TEMP_AREA, S_ADR                ; 8408
                                56            0D  D0 00147           MOVL    #13, D_LEN                      ; 8409
                                              6E  11 0014A 14$:      BRB     23$                             ; 8402
                                26      04    AE  D1 0014C 15$:      CMPL    IDENT_LEN, #38                  ; 8412
                                              40  11 00150           BRB     21$
                                56            08  D0 00152 16$:      MOVL    #8, D_LEN                       ; 8420
                                50      26    A8  9E 00155           MOVAB   38(R8), R0                      ; 8421
                                57            50  D0 00159           MOVL    R0, D_ADR
                                01      07    A9  91 0015C           CMPB    7(R9), #1                       ; 8422
                                              19  12 00160           BNEQ    18$
                                57            50  D0 00162           MOVL    R0, D_ADR                       ; 8425
                                        10    AE  9F 00165           PUSHAB  TEMP_AREA                       ; 8426
                                              5A  DD 00168           PUSHL   S_ADR
                            FE43    CF            02  FB 0016A           CALLS   #2, TO_ODS1_DATE
                                5B            0D  D0 0016F           MOVL    #13, S_LEN                      ; 8427
                                5A      10    AE  9E 00172           MOVAB   TEMP_AREA, S_ADR                ; 8428
                                56            07  D0 00176           MOVL    #7, D_LEN                       ; 8429
                                              75  11 00179 17$:      BRB     30$                             ; 8422
                                2E      04    AE  D1 0017B 18$:      CMPL    IDENT_LEN, #46                  ; 8432
                                              11  11 0017F 19$:      BRB     21$
                                56            08  D0 00181 20$:      MOVL    #8, D_LEN                       ; 8440
                                57      2E    A8  9E 00184           MOVAB   46(R8), D_ADR                   ; 8441
                                01      07    A9  91 00188           CMPB    7(R9), #1                       ; 8442
                                              60  13 0018C           BEQL    29$
                                36      04    AE  D1 0018E           CMPL    IDENT_LEN, #54                  ; 8446
                                              7E  1E 00192 21$:      BGEQU   36$
                                              58  11 00194           BRB     29$                             ; 8448
                                56            04  D0 00196 22$:      MOVL    #4, D_LEN                       ; 8454
```

```
                        57      3C  A9  9E 00199          MOVAB   60(R9), D_ADR                    ; 8455
                        01      07  A9  91 0019D          CMPB    7(R9), #1                       ; 8456
                                    6F  12 001A1          BNEQ    36$                             ;
                    10  AE          6A  90 001A3          MOVB    (S_ADR), TEMP_AREA              ; 8459
                    11  AE      02  AA  90 001A7          MOVB    2(S_ADR), TEMP_AREA+1           ; 8460
                        5B          02  D0 001AC          MOVL    #2, S_LEN                       ; 8461
                        5A      10  AE  9E 001AF          MOVAB   TEMP_AREA, S_ADR                ; 8462
                        56          02  D0 001B3          MOVL    #2, D_LEN                       ; 8463
                        57      08  A9  9E 001B6          MOVAB   8(R9), D_ADR                    ; 8464
                                    78  11 001BA 23$:      BRB     39$                             ; 8296
                        56          02  D0 001BC 24$:      MOVL    #2, D_LEN                       ; 8471
                        57      40  A9  9E 001BF          MOVAB   64(R9), D_ADR                   ; 8472
                        01      07  A9  91 001C3          CMPB    7(R9), #1                       ; 8473
                                    6B  12 001C7          BNEQ    39$                             ;
                        57      0A  A9  9E 001C9          MOVAB   10(R9), D_ADR                   ; 8475
                                    65  11 001CD          BRB     39$                             ; 8296
                        56          02  D0 001CF 25$:      MOVL    #2, D_LEN                       ; 8481
                        57      38  A9  9E 001D2          MOVAB   56(R9), D_ADR                   ; 8482
                                    10  11 001D6          BRB     28$                             ; 8483
                        56          02  D0 001D8 26$:      MOVL    #2, D_LEN                       ; 8491
                        57      48  A9  9E 001DB          MOVAB   72(R9), D_ADR                   ; 8492
                                    07  11 001DF          BRB     28$                             ; 8493
                        56          01  D0 001E1 27$:      MOVL    #1, D_LEN                       ; 8501
                        57      3B  A9  9E 001E4          MOVAB   59(R9), D_ADR                   ; 8502
                        01      07  A9  91 001E8 28$:      CMPB    7(R9), #1                       ; 8503
                                    46  12 001EC          BNEQ    39$                             ;
                                    56  D4 001EE 29$:      CLRL    D_LEN                           ; 8505
                                    42  11 001F0 30$:      BRB     39$                             ; 8296
                        02      07  A9  91 001F2 31$:      CMPB    7(R9), #2                       ; 8511
                                    0F  12 001F6          BNEQ    34$                             ;
                        56          06  D0 001F8          MOVL    #6, D_LEN                       ; 8513
                        57      0E  A9  9E 001FB 32$:      MOVAB   14(R9), D_ADR                   ; 8514
                                    33  11 001FF          BRB     39$                             ; 8296
                        02      07  A9  91 00201 33$:      CMPB    7(R9), #2                       ; 8520
                                    04  13 00205          BEQL    35$                             ;
                        50          34  D0 00207 34$:      MOVL    #52, R0                         ; 8521
                                        04 0020A          RET                                     ;
                        56          02  D0 0020B 35$:      MOVL    #2, D_LEN                       ; 8522
                        57      04  A9  9E 0020E          MOVAB   4(R9), D_ADR                    ; 8523
                                    20  11 00212 36$:      BRB     39$                             ; 8296
                        02      07  A9  91 00214 37$:      CMPB    7(R9), #2                       ; 8530
                                    17  12 00218          BNEQ    38$                             ;
                                0C  AC  DD 0021A          PUSHL   FIB                             ; 8532
                52          0C  AE      04  C1 0021D      ADDL3   #4, ATR, R2                     ;
                                    62  DD 00222          PUSHL   (R2)                            ;
                        7E      10  BE  3C 00224          MOVZWL  @ATR, -(SP)                     ; 8531
                                    51  DD 00228          PUSHL   R1                              ;
                    00000000G       00  04  FB 0022A      CALLS   #4, ACL_DISPATCH                ;
                                    6E  01  D0 00231 38$:      MOVL    #1, ACL_ATR                ; 8533
                                    06  6E  E8 00234 39$:      BLBS    ACL_ATR, 40$               ; 8544
            56      00              6A  5B  2C 00237          MOVC5   S_LEN, (S_ADR), #0, D_LEN, (D_ADR)  ;
                                        67  0023C          ;
                            08  AE      08  C0 0023D 40$:      ADDL2   #8, ATR                    ; 8545
                                FDDD    31  00241          BRW     1$                              ; 8280
                        50          01  D0 00244 41$:      MOVL    #1, R0                          ; 8550
                                        04 00247          RET                                     ;
```

; Routine Size: 584 bytes, Routine Base: CODE + 39D3

```
; 7046        8551  1 END
; 7047        8552  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY

;       Name                    Bytes                   Attributes

;  COMMON                        2124  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;  DATA                            48  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  CODE                         15387  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                           Library Statistics

;                                    -------- Symbols --------      Pages       Processing
;       File                         Total    Loaded   Percent      Mapped      Time

;  _$255$DUA28:[SYSLIB]LIB.L32;1      18619       328         1      1000          00:02.2


;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:STAACP/OBJ=OBJ$:STAACP MSRC$:STAACP/UPDATE=(ENH$:STAACP)

; Size:          15119 code + 2440 data bytes
; Run Time:          04:26.5
; Elapsed Time:      14:37.3
; Lines/CPU Min:      1925
; Lexemes/CPU-Min: 21788
; Memory Used:  802 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

STAACP
LIS

STAACCTL
LIS

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

STABACCOP
LIS

STAINIT
LIS